Web aplikacija za kreiranje narudžbe u restoranu

Kolarić, Toni

Undergraduate thesis / Završni rad

2023

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

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:977472

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-15**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Stručni studij informatike

WEB APLIKACIJA ZA KREIRANJE NARUDŽBE U RESTORANU

Završni rad

Toni Kolarić

Osijek, 2023.

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

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

Osijek, 05.07.2023.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za završni ispit na preddiplomskom stručnom studiju

Ime i prezime Pristupnika:	Toni Kolarić
Studij, smjer:	Preddiplomski stručni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	AR4670, 26.07.2018.
OIB Pristupnika:	77467069117
Mentor:	Robert Šojo, mag. ing. comp.
Sumentor:	,
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	mr. sc. Željko Štanfel
Član Povjerenstva 1:	Robert Šojo, mag. ing. comp.
Član Povjerenstva 2:	Marina Peko, dipl. ing.
Naslov završnog rada:	Web aplikacija za kreiranje narudžbe u restoranu
Znanstvena grana završnog rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak završnog rada	Aplikacija omogućava kreiranje narudžbe putem QR kôda za stolom gdje se sjedi. Svakom narudžbom koordiniraju se konobari s kuhinjom i posluživanjem pića i hrane gostu. Unutar aplikacije postoji administrator, zaposlenik u ulozi konobara i konzument aplikacije koji je gost u restoranu. Student treba razviti funkcionalnu aplikaciju koristeći različite web tehnologije. Rezervirano za studenta: Toni Kolarić
Prijedlog ocjene pismenog dijela ispita (završnog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	05.07.2023.

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

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

IZJAVA O ORIGINALNOSTI RADA

Osijek, 13.07.2023.

Ime i prezime studenta:	Toni Kolarić
Studij:	Preddiplomski stručni studij Računarstvo
Mat. br. studenta, godina upisa:	AR4670, 26.07.2018.
Turnitin podudaranje [%]:	7

Ovom izjavom izjavljujem da je rad pod nazivom**: Web aplikacija za kreiranje narudžbe u restoranu**

izrađen pod vodstvom mentora Robert Šojo, mag. ing. comp.

i sumentora ,

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

Potpis studenta:

SADRŽAJ

1. UVOD 1
1.1. Zadatak završnog rada 1
2. POSTOJEĆE APLIKACIJE 2
2.1. Menufy aplikacija 2
2.2. GrubHub aplikacija 3
2.3. Waiterio aplikacija 3
3. WEB APLIKACIJA 5
3.1. Prednosti web aplikacije 5
4. KORIŠTENE TEHNOLOGIJE PRILIKOM IZRADE WEB APLIKACIJE 7
4.1. HTML 8
4.2. CSS 9
4.3. Javascript 10
4.4. Bootstrap 11
4.5. Vue.js 12
4.5.1. Element Plus - Vue.js components package 13
4.5.2. Pinia - Vue.js store package 14
4.5.3. Axios - biblioteka 15
4.6. PHP 16
4.7. SQL 17
4.7.1. MySQL 18
4.8. Visual Studio Code 19
5. IZRADA APLIKACIJE 20
5.1. Baza podataka 21
5.2. Spremanje narudžbi - REST API 25
5.3. Prijava u aplikaciju 26
5.4. Stvaranje i brisanje računa 27
5.4.1. Stvaranje računa 27
5.4.2. Brisanje računa 28
5.5. Stvaranje, brisanje i ažuriranje artikla 28
5.5.1. Stvaranje artikla 28
5.5.2. Ažuriranje artikla 30
5.5.3. Brisanje artikla 31
5.6. Statistički prikaz artikala 31
5.7. Generator slučajnih četveroznamenkastih brojeva 32
5.8. Logika naručivanja 32
5.9. Generiranje QR kodova 33

6. KORIŠTENJE APLIKACIJE 34
6.1. Korisnik - gost u restoranu 34
6.2. Konobar 35
6.3. Kuhar 36
6.4. Administrator 36
6.5. Generator 37
7. ZAKLJUČAK 39
LITERATURA 41
SAŽETAK 42
ABSTRACT 43
PRILOZI 44
ŽIVOTOPIS 45

1. UVOD

Danas je sve više restorana koji se odlučuju na implementaciju modernih tehnologija kako bi poboljšali korisničko iskustvo i olakšali poslovanje. Jedna od takvih tehnologija je web aplikacija za narudžbu unutar restorana putem QR koda. Ova tehnologija omogućuje korisnicima jednostavnu narudžbu hrane i pića putem svog mobilnog uređaja, čime se značajno smanjuju čekanja i gužve u restoranu. Uz to, konobari i kuhari dobivaju narudžbu direktno na svoje uređaje, što smanjuje vrijeme čekanja na obradu narudžbe. Ova tehnologija također omogućuje korisnicima da pristupe svojoj košarici i naruče novu stavku ili zatraže plaćanje bez potrebe za pozivanjem konobara. Uz sve to, ovaj način narudžbe u restoranu je i ekološki prihvatljiv, s obzirom da se smanjuje potrošnja papira za menije i narudžbe. U ovom završnom radu detaljno je opisano kako je razvijena web aplikaciju za narudžbu unutar restorana putem QR koda, kao i prednosti i nedostatke ove tehnologije.

U drugom poglavlju napravljen je pregled postojećih aplikacija kao i kratki osvrt na njih te njihova usporedba sa postojećom. Trećim poglavljem je ukratko opisana web aplikacija, kako web aplikacija radi te su navedene njezine prednosti. U četvrtom poglavlju navedene su sve tehnologije i programski jezici koji su korišteni prilikom izrade, te su ukratko opisane i objašnjene njihove osnovne funkcije. Peto poglavlje opisuje izradu aplikacije te svih njezinih dijelova od koji se sastoji, na koji se način ti dijelovi koriste, gdje se nalaze i na koji način rade. U šestom poglavlju su opisani načini autorizacije korisnika te njihove uloge i mogućnosti.

1.1. Zadatak završnog rada

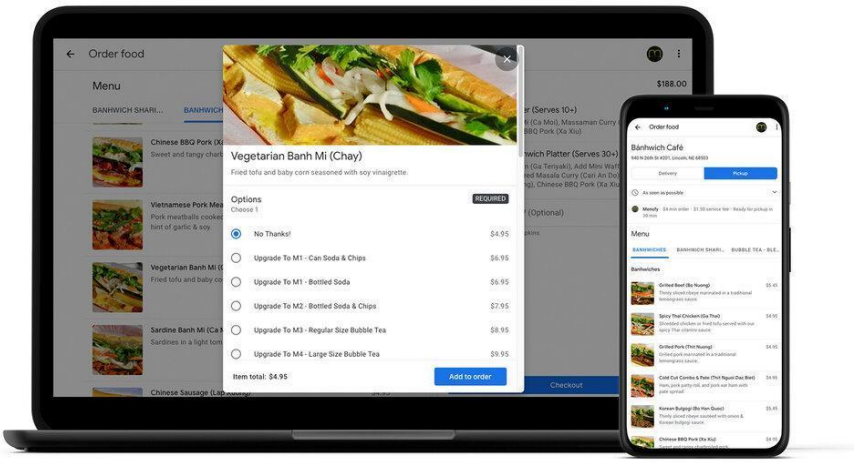
Aplikacija omogućava kreiranje narudžbe putem QR koda za stolom gdje se sjedi. Svakom narudžbom koordiniraju se konobari s kuhinjom i posluživanjem pića i hrane gostu. Unutar aplikacije postoji administrator, zaposlenik u ulozi konobara i konzument aplikacije koji je gost u restoranu.

2. POSTOJEĆE APLIKACIJE

U ovom poglavlju su opisane slične aplikacije koje koriste tehnologiju QR koda i olakšavaju korisnicima proces naručivanja hrane i pića u restoranima. Također, navedene su njihove prednosti i nedostaci.

2.1. Menufy aplikacija

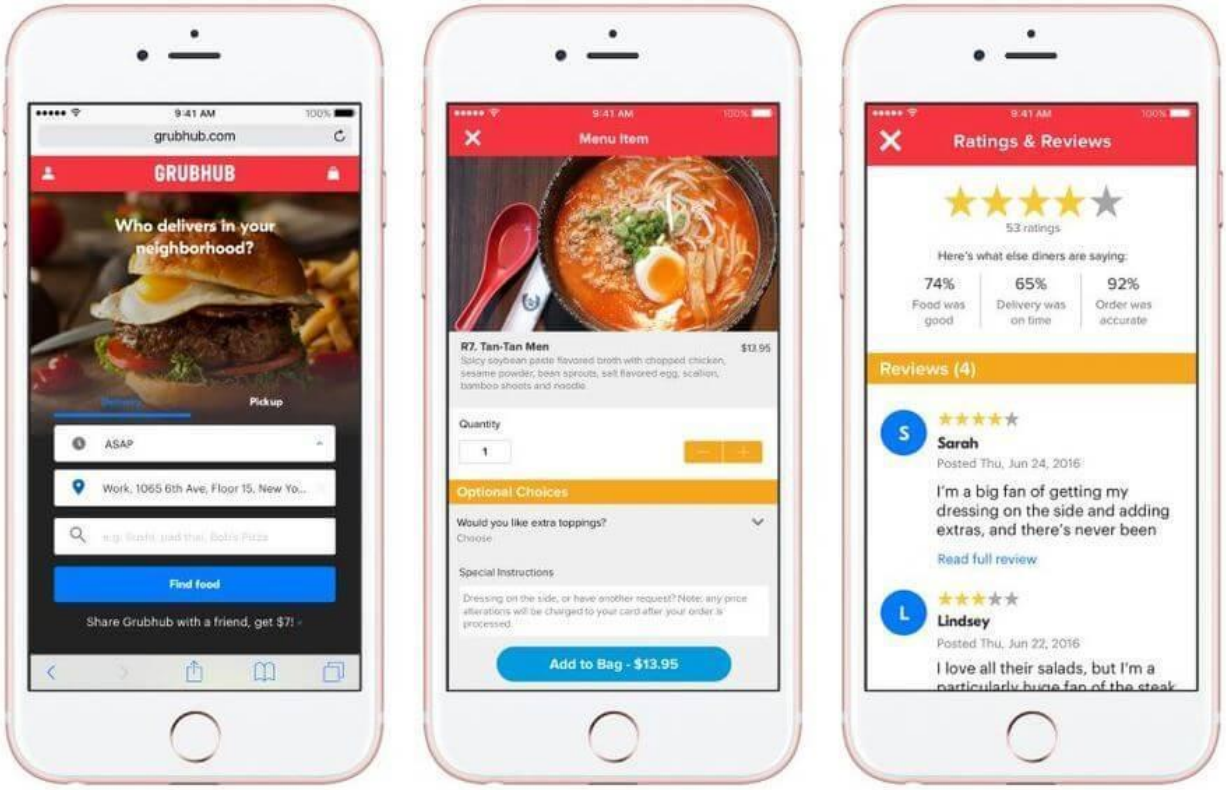
Jedna od sličnih aplikacija je Menufy koja omogućuje korisnicima skeniranje QR koda kako bi pristupili jelovniku i napravili narudžbu putem mobilnog uređaja [1]. Konobari dobivaju narudžbu na svojim uređajima, a korisnici mogu pratiti status narudžbe i očekivano vrijeme dostave. Prednost ove aplikacije je jednostavnost korištenja i mogućnost narudžbe iz udobnosti stola, dok nedostatak može biti veća mogućnost pogrešnog unosa narudžbe od strane korisnika izazvane korisničkim sučeljem (slika 2.1).



Slika 2.1. *Menufy - stranica za naručivanje.*

2.2. GrubHub aplikacija

Druga slična aplikacija je GrubHub koja također omogućuje korisnicima naručivanje hrane putem QR koda [2]. Ova aplikacija ima širi izbor restorana nego što je to slučaj s Menufy, ali ne nudi uvijek jednostavno i brzo naručivanje u samom restoranu. Prednost ove aplikacije je širok izbor restorana i hrane, dok nedostatak može biti čekanje na dostavu (slika 2.2).



Slika 2.2. *GrubHub aplikacija.*

2.3. Waiterio aplikacija

Waiterio je aplikacija za naručivanje hrane u restoranima i kafićima koja koristi QR kodove [3]. Gosti mogu skenirati QR kod na svom stolu za pristup digitalnom jelovniku, pregledu dostupnih jela i pića, te izravno naručiti iz aplikacije.

Osim toga, Waiterio omogućuje plaćanje računa putem aplikacije, što olakšava cjelokupno iskustvo za goste. Aplikacija ne samo da smanjuje mogućnost grešaka u narudžbama, već i poboljšava učinkovitost restorana omogućujući osoblju da se fokusira na pripremu jela i pružanje bolje usluge.

S intuitivnim korisničkim sučeljem, Waiterio omogućuje gostima da naruče obroke u samo nekoliko dodira. Ukupno gledano, Waiterio je inovativna aplikacija koja koristi digitalnu tehnologiju za poboljšanje iskustva naručivanja hrane u restoranima i kafićima.

3. WEB APLIKACIJA

Web aplikacija je inovativni računalni program koji se koristi putem web preglednika i web tehnologije. Ova tehnologija omogućuje korisnicima pristup nizu rješenja putem interneta. Funkcionira na principu klijent-server arhitekture, gdje se klijentu pruža grafičko sučelje koje je definirano na serveru. Upravljanje pohranom i dohvaćanjem informacija koriste kombinaciju skripti na strani poslužitelja PHP (engl. *Hypertext Preprocessor*) i ASP (engl. *Active Server Pages*), dok se skripte na strani klijenta JavaScript i HTML (engl. *HyperText Markup Language*) koriste za prezentiranje informacija korisnicima.

Prilikom izrade ovog projekta koristio se Vue.js 3 tehnologija koja koristi Javascript, HTML i CSS, a za povezivanje s MySQL bazom podataka PHP. Vue 3 tehnologija je moderna i popularna tehnologija za izradu dinamičkih web aplikacija. Vue 3 omogućuje lakši razvoj i održavanje aplikacija zahvaljujući fleksibilnoj arhitekturi i jednostavnoj sintaksi. Pozadinski dio aplikacije koristi PHP kao programski jezik za spajanje na MySQL bazu podataka, što omogućuje spremanje podataka i njihovo dohvaćanje iz baze. Korištenje PHP-a u kombinaciji s MySQL bazom podataka često se koristi u razvoju web aplikacija zbog svoje popularnosti i dostupnosti. Aplikacija izvrsno kombinira ove tehnologije za postizanje funkcionalnosti i performansi.

3.1. Prednosti web aplikacije

Web aplikacije imaju brojne prednosti u odnosu na tradicionalne aplikacije koje se preuzimaju i instaliraju na uređaje. Web aplikacije se ne moraju preuzeti, a korisnicima su dostupne putem web preglednika na svim uređajima koji imaju pristup internetu. Također, korisniku nije potrebno ažurirati aplikaciju na svom uređaju, jer se web aplikacija ažurira na poslužitelju. Osim toga, web aplikacije su pristupačne na svim platformama i operativnim sustavima, što ih čini vrlo fleksibilnima i dostupnima korisnicima.

Jedna od novijih tehnologija u svijetu web aplikacija su progresivne web aplikacije, odnosno PWA (enlg. *Progressive web apps*). PWA kombinira prednosti web aplikacija i nativnih aplikacija, čime korisnicima pruža vrhunsko korisničko iskustvo. PWA-ovi se mogu koristiti na svim uređajima

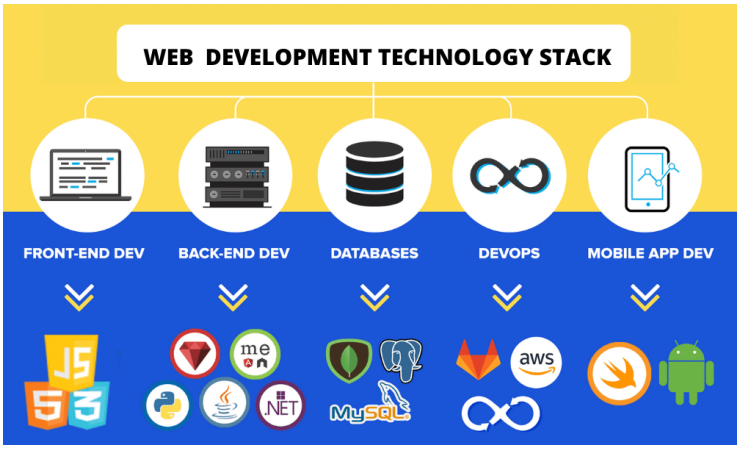
i operativnim sustavima, a omogućuju pristup različitim funkcijama uređaja kao što su kamera, GPS, senzori pokreta i drugo. PWA-ovi se mogu instalirati na uređaj kao nativna aplikacija, što omogućuje brži pristup i korisničko iskustvo slično kao kod nativnih aplikacija [4].

Ukratko, web aplikacije nude brojne prednosti u odnosu na tradicionalne aplikacije, a PWA tehnologija pruža korisnicima još naprednije mogućnosti i iskustvo korištenja. Stoga je izrada web aplikacija jedan od ključnih trendova u razvoju modernih aplikacija i tehnologija koja će vjerojatno imati sve veću važnost u budućnosti.

4. KORIŠTENE TEHNOLOGIJE PRILIKOM IZRADE WEB APLIKACIJE

Web aplikacije su danas postale neizostavan dio naših života, bilo da se radi o jednostavnim blogovima, društvenim mrežama ili složenim poslovnim aplikacijama. Razvoj web aplikacija zahtijeva kombinaciju različitih tehnologija koje omogućuju stvaranje funkcionalne i privlačne aplikacije koja će zadovoljiti potrebe korisnika.

Jedna od osnovnih podjela u razvoju web aplikacija je podjela na backend, frontend i bazu podataka (slika 4.1). Backend je dio aplikacije koji se odnosi na sve procese koji se odvijaju na poslužiteljskoj strani. Ovdje se nalazi gotovo sva logika i algoritmi koji obrađuju podatke koje je korisnik poslao i vraćaju mu obrađene podatke. Backend obično koristi programski jezik kao što su Python, Java, Ruby, PHP i drugi, a koristi i različite radne okoline i alate poput Node.js, Flask, Django, Rails i drugih.



Slika. 4.1. *Tehnologije za izradu aplikacija [5].*

Frontend, s druge strane, odnosi se na sve procese koji se odvijaju na korisničkoj strani aplikacije. Ovdje se nalaze svi elementi koji se vide i s kojima korisnik interaktira. Frontend koristi

tehnologije kao što su HTML, CSS i JavaScript, a koristi i različite biblioteke i okvire kao što su React, Angular, Vue i drugi. Cilj frontend dijela aplikacije je da korisniku pruži intuitivan i ugodan korisnički doživljaj i omogući mu da obavlja željene funkcionalnosti na jednostavan način (služi za što bolje UX (engl. *user experience*) i UI (engl. *user interface*) korisničko iskustvo).

Baza podataka je dio aplikacije koji se koristi za pohranu, organizaciju i upravljanje podacima. Baza podataka je ključna za sve web aplikacije koje obrađuju veliku količinu podataka i gdje je važno da podaci budu pohranjeni na siguran i organiziran način. Najčešće se koriste relacijske baze podataka kao što su MySQL, PostgreSQL, Oracle i druge, ali se u novije vrijeme sve više koriste i različite vrste nerelacijskih baza podataka kao što su MongoDB, Cassandra, Redis i drugi.

Pri razvoju web aplikacija, važno je da se backend, frontend i baza podataka međusobno dobro integriraju kako bi aplikacija bila funkcionalna, brza i sigurna. Korištenje modernih tehnologija i alata, kao i poznavanje njihovih prednosti i nedostataka, ključno je za uspješan razvoj kvalitetne web aplikacije koja će zadovoljiti potrebe korisnika i poslovanja.

4.1. HTML

HTML je standardni jezik koji se koristi za stvaranje i dizajniranje web stranica. To je osnovni jezik za stvaranje i prikazivanje web sadržaja na svim internetskim preglednicima [6].

HTML se sastoji od niza oznaka koje se koriste za opisivanje strukture, sadržaja i stilova web stranica. Oznake se koriste za definiranje različitih elemenata na web stranici, kao što su tekst, slike, tabele, forme i drugi elementi. Također se koriste za stvaranje veza između web stranica. Neprestano se razvija i ažurira kako bi se poboljšale mogućnosti izrade web stranica. Nova izdanja HTML-a donose nove elemente i mogućnosti, što olakšava stvaranje modernih web stranica. Trenutno najnovije izdanje HTML-a je HTML5, koje nudi bogat skup novih elemenata i poboljšanja. Jedna od važnih karakteristika HTML-a je njegova lakoća korištenja. Kod je relativno jednostavan za razumijevanje i čitanje, što ga čini pristupačnim i za početnike u web razvoju. Osim toga, HTML je jezik koji se može koristiti na svim operacijskim sustavima što ga čini vrlo fleksibilnim i pristupačnim.

Nerijetko se koristi zajedno s drugim jezicima i tehnologijama za stvaranje modernih web aplikacija. Jedna od popularnih tehnologija koja se koristi s HTML-om je CSS koja se koristi za

stiliziranje web stranica i definiranje izgleda elemenata. Također se često koristi i JavaScript, koji se koristi za dinamičko ažuriranje web stranica i stvaranje interaktivnih elemenata.

Uzimajući u obzir sve prednosti HTML-a, jasno je zašto je ovaj jezik i dalje važan u razvoju web aplikacija i dizajnu web stranica. To je osnovni jezik koji se koristi prilikom web razvoja i ključan je za stvaranje bilo koje vrste web sadržaja. HTML ostaje jednostavan za korištenje, pristupačan i fleksibilan, što ga čini vrlo važnim alatom u svijetu web razvoja (slika 4.2).

```
<form action="/login" method="post">
 <label for="username">Korisničko ime:</label>
 <input type="text" id="username" name="username" required>

 <label for="password">Lozinka:</label>
 <input type="password" id="password" name="password" required>

 <button type="submit">Prijavi se</button>
</form>
```

Slika 4.2. *Primjer HTML forme za prijavu korisnika.*

4.2. CSS

CSS (engl. *Cascading Style Sheets*) je stilski jezik koji se koristi za opisivanje izgleda i stila web stranica. CSS se koristi za definiranje boja, fontova, pozadina, rasporeda, animacija i drugih vizualnih elemenata web stranice (slika 4.3) [7].

Prednosti korištenja CSS-a su mnogobrojne. Prije svega, CSS odvaja stilove od sadržaja, što omogućava lakše održavanje i ažuriranje web stranice. Također, CSS omogućava brže učitavanje web stranice jer se stilovi učitavaju u zasebnoj datoteci, što znači da se sadržaj web stranice može brže učitati.

CSS omogućava i responzivnost web stranice, što znači da se stranica automatski prilagođava veličini zaslona na kojem se prikazuje. Ovo je posebno važno u današnjem mobilnom svijetu prilikom prikaza na manjim uređajima.

Olakšava korištenje složenih i ponavljajućih stilova. Na primjer, može se definirati stil za zaglavlja na cijeloj web stranici, a zatim taj stil primijeniti na svaki naslov na web stranici. Ovo je vrlo korisno za održavanje dosljednosti u izgledu i stilu web stranice.

S vremenom je postao standardni jezik koji se koristi prilikom web dizajnu i podržava ga gotovo svaki moderniji web preglednik. Također postoji mnogo resursa i alata za učenje i korištenje CSS-a, što olakšava razvoj i održavanje web stranica.

Korištenje CSS-a je neophodno za moderno web dizajniranje. On omogućava web developerima precizno definiranje stila i izgleda web stranice, što je ključno za privlačenje korisnika i izgradnju uspješnog dizajna web aplikacije.

```
<style>
  .text-red {
    color: red;
  }

  .bg-blue {
    background-color: blue;
  }

  .border-green {
    border: 2px solid green;
  }
</style>
```

Slika 4.3. *Primjer definiranja osnovnih CSS klasa.*

4.3. Javascript

JavaScript je popularan programski jezik koji se široko koristi u web razvoju. Kao skriptni jezik, JavaScript se koristi za stvaranje interaktivnih i dinamičkih web stranica koje se lako mogu ažurirati i prilagoditi. Osim toga, JavaScript se može koristiti i za razvoj mobilnih aplikacija, desktop aplikacija i igara [8].

Jedna od glavnih prednosti JavaScripta je njegova sposobnost izvršavanja na klijentskoj strani, što znači da se kod izvršava na samom uređaju korisnika, bez potrebe za stalnom komunikacijom s poslužiteljem. Ovo omogućava brz i glatki prikaz web stranica, čime se poboljšava korisničko iskustvo.

JavaScript se koristi za rješavanje različitih vrsta zadataka prilikom web razvoja, uključujući validaciju podataka u formama, izradu interaktivnih animacija i igara, ažuriranje dinamičkog sadržaja, stvaranje grafičkih efekata i još mnogo toga. Također podržava integraciju s drugim tehnologijama i jezicima, čime se omogućava stvaranje kompleksnih web stranica.

JavaScript je relativno jednostavan za naučiti, zbog svoje čitljivosti i sintakse koja nalikuje na ostale programski jezike. Postoje mnogi besplatni resursi za učenje JavaScripta, kao što su online tečajevi, knjige i tutorijali.

Jedan od trendova u web razvoju koji koristi JavaScript je Progressive Web App (PWA), koji omogućava korisnicima da koriste web aplikacije kao da su instalirane na njihovim uređajima. Ovo se postiže kombinacijom web tehnologija i funkcionalnosti koje se obično povezuju s mobilnim aplikacijama, kao što su brzo učitavanje, offline funkcionalnost i obavijesti.

Uzimajući u obzir sve prednosti JavaScripta, jasno je zašto je ovaj jezik toliko važan u svijetu web razvoja. JavaScript se koristi za stvaranje modernih i interaktivnih web stranica i aplikacija koje su prilagođene korisničkim potrebama. Kao pristupačan i fleksibilan programski jezik, JavaScript omogućava web developerima da stvore bogatiji i dinamičniji web sadržaj.

4.4. Bootstrap

Bootstrap je popularni CSS framework koji se koristi za brzu i jednostavnu izradu responzivnih web stranica i web aplikacija. Razvijen je od Twittera, a kasnije je postao projekt otvorenog koda [9].

Jedna od najvećih prednosti Bootstrapa je u tome što dolazi s predlošcima za mnoge elemente web stranice, kao što su gumbi, forme, navigacijske trake, kartice, tipografija i još mnogo toga. Ovi predlošci dolaze s predefiniranim CSS stilovima koji se mogu prilagoditi kako bi se postigao željeni izgled web stranice (slika 4.4).

Bootstrap također sadrži veliki broj JavaScript komponenti, kao što su modalni prozori, tabovi, skočni opisi, animacije i još mnogo toga. Ove komponente omogućavaju web programerima lako dodavanje interaktivnost i funkcionalnost na web stranicu.

Bootstrap se također brine o responzivnosti web stranice, što znači da se stranica automatski prilagođava veličini zaslona na kojem se prikazuje. Ovo je posebno korisno u današnjem svijetu mobilnih uređaja, gdje se web stranice često pregledavaju na manjim zaslonima.

Korištenje Bootstrapa olakšava razvoj i održavanje web stranica. Kako se Bootstrap koristi za razvoj responzivnih web stranica, developeri se ne moraju brinuti o tome kako će se stranica prikazati na različitim uređajima. Osim toga, Bootstrap ima veliku zajednicu korisnika koji nude pomoć i savjete, kao i mnoge resurse koji pomažu prilikom učenja i korištenja Bootstrapa.

S obzirom na sve ove prednosti, jasno je zašto Bootstrap ostaje popularan izbor širom svijeta. On omogućava brži i jednostavniji razvoj responzivnih web stranica s modernim i atraktivnim dizajnom.

```
<div class="container">
  <div class="row">
    <div class="col-sm-12">
      <h1>Dobrodošli na našu web stranicu</h1>
    </div>
  </div>
</div>
```

Slika 4.4. *Primjer primjene Bootstrap klasa.*

4.5. Vue.js

Vue.js je tehnologija otvorenog programskog koda (engl. *open-source*), te progresivni JavaScript okvir koji se koristi za izgradnju korisničkih sučelja i najčešće jednostranih web aplikacija. Ovaj okvir je napravljen s ciljem olakšavanja razvoja modernih aplikacija koje su skalabilne i lakše održive (slika 4.5). Vue.js je vrlo popularan prilikom razvoja web aplikacija zbog svoje jednostavnosti, fleksibilnosti i brzine [10].

Jedna od najvažnijih karakteristika Vue.jsa je njegova arhitektura komponenata. Aplikacije se sastoje od više malih i nezavisnih komponenti koje se mogu lako kombinirati i ponovo koristiti u različitim dijelovima aplikacije. Ovo olakšava razvoj i održavanje aplikacije jer se fokusira na odvojenost i veću iskoristivost komponenti.

Vue.js također koristi virtualni DOM (engl. *Document Object Model*) koji omogućuje brže ažuriranje korisničkog sučelja. Virtualni DOM je sličan pravom DOM-u, ali Vue.js ga koristi za ažuriranje samo onih dijelova korisničkog sučelja koji se mijenjaju. Ovo ubrzava proces ažuriranja i smanjuje potrošnju resursa.

Vue.js ima vrlo aktivnu zajednicu koja kontinuirano doprinosi razvoju i unapređivanju tehnologije. Također postoje i različiti dodaci i biblioteke koje se mogu koristiti s Vue.jsom, kao što su Vuex za upravljanje stanjima, Vue Router za upravljanje rutama u jednostranim aplikacijama, Element Plus za integriranje gotovih pojednostavljenih komponenti, Pinia kao pohranjivanje i mnoge druge.

Prilikom izrade web aplikacije pomoću Vue.jsa, mogu se koristiti i različiti dodatni alati kao što su Vue CLI (engl. *Command Line Interface*) za automatizaciju postavljanja projekta i Webpack za upravljanje paketima i modulima. Sve ove karakteristike omogućuju programerima brži i efikasniji razvoj modernih web aplikacija.

Uzimajući u obzir prednosti Vue.jsa, nije iznenađujuće što se ovaj okvir često koristi za razvoj jednostranih aplikacija i korisničkih sučelja. Vue.js nudi fleksibilnost i jednostavnost, a istovremeno omogućava razvoj skalabilnih aplikacija koje se mogu lako održavati. Ova tehnologija se uvelike koristi u industriji, a njena popularnost raste iz godine u godinu.

```
<template>
 <div>
   <h1>{{ naslov }}</h1>
   <p>{{ poruka }}</p>
 </div>
</template>

<script>
export default {
 name: 'MojComponent',
 data() {
   return {
     naslov: 'Dobrodošli na moju Vue.js stranicu',
     poruka: 'Ovo je primjer Vue.js datoteke'
   }
 }
}
</script>

<style>
 h1 {
  font-size: 2rem;
```

```
 color: blue;
}
p {
 font-size: 1.5rem;
 color: green;
}
</style>
```

Slika 4.5. *Primjer jednostavne Vue.js datoteke.*

4.5.1. Element Plus - Vue.js components package

Element Plus je dodatak koji proširuje osnovni set UI komponenti dostupnih u Vue.js-u. Temelji se na Element UI i dodaje nove, poboljšane, jednostavne i optimiziranije komponente.

Pruža bogat skup korisničkih sučelja koji olakšavaju izgradnju modernog i reaktivnog web dizajna. Neki od najpopularnijih komponenti Element Plusa uključuju tablice, forme, gumbe, izbornike, selektore datuma, okvire i još mnogo drugih.

Element Plus uključuje nekoliko naprednih značajki, kao što su optimizacija performansi, podrška za internacionalizaciju, pretraživanje i filtriranje u tablicama, reaktivno ažuriranje podataka itd.

Osim svojih unaprijeđenih komponenti, on ima jednostavnu i intuitivnu dokumentaciju i aktivnu zajednicu koja pruža podršku i održavanje. Element Plus je jednostavan za korištenje i olakšava proces izrade UI-a za vašu Vue.js aplikaciju (slika 4.6).

Sve u svemu, Element Plus je moćan dodatak za Vue.js koji proširuje osnovni set komponenti dostupnih u samoj Vue.js tehnologiji. Sa svojim naprednim značajkama definitivno poboljšava i olakšava izradu modernih i reaktivnih web aplikacija [11].

```
<template>
 <div class="app">
  <el-input
   v-model="textValue"
   placeholder="Unesite tekst ovdje"
```

```
    @input="handleInput" />
 </div>
</template>
<script>
export default {
 data() {
   return {
     textValue: '',
   }
 },
 methods: {
   handleInput(value) {
     console.log('Unesena vrijednost:', value)
   },
 },
}
</script>
<style scoped>
.app {
 margin: 20px;
}
</style>
```

Slika 4.6. *Primjer 'input' komponente u Vue 3 tehnologiji koristeći Element Plus paket.*

4.5.2. Pinia - Vue.js store package

Pinia je novi vuex alternativni upravitelj stanja (engl. *state management*) sustav za Vue.js aplikacije koji se temelji na Vue 3 reaktivnosti. To je lagan i brz sustav koji olakšava upravljanje globalnim stanjem u Vue.js aplikaciji [12].

Sastoji se od dvije glavne komponente: store i dodatak. Store je objekt koji sadrži globalno stanje vaše aplikacije, a dodatak je Vue.js dodatak koji vam omogućuje integraciju Pinia storea u vašu aplikaciju.

Svojom jednostavnošću i performansama, Pinia je postala popularna opcija za upravljanje stanjem u Vue.js aplikacijama. Ona se temelji na Vue 3 reaktivnosti, što znači da je vrlo reaktivna i brza za korištenje.

Pinia ima vrlo jednostavnu i intuitivnu dokumentaciju, koja je lako razumljiva i sadrži dobre primjere koda. Također ima veliku zajednicu koja pruža podršku i održavanje, što je čini pouzdanom opcijom za upravljanje stanjima u vašoj Vue.js aplikaciji.

Korištenje Pinia u aplikaciji omogućuje upravljanje globalnim stanjem na jednostavan i intuitivan način, te svojim performansama, jednostavnošću i aktivnom zajednicom, postala je vrlo popularna opcija.

4.5.3. Axios - biblioteka

Axios je popularna JavaScript biblioteka koja omogućava izvođenje asinkronih HTTP zahtjeva u web aplikacijama. U ovoj aplikaciji koristi se Axios kao glavna biblioteka za upravljanje zahtjevima na frontendu Vue 3 aplikacije [13].

Korištenje Axiosa u Vue 3 je vrlo jednostavno (slika 4.7), jer se Axios može instalirati kao plugin u Vue aplikaciju. Nakon instalacije, Axios je dostupan u cijeloj aplikaciji kroz Vue instancu. To znači da se Axios može koristiti za slanje zahtjeva na API, prikupljanje podataka i ažuriranje aplikacije.

U projektu ovog završnog rada Axios se koristi za slanje zahtjeva na REST API za narudžbe hrane i pića. Kada korisnik potvrdi svoju narudžbu, Axios je koristio HTTP POST metodu za slanje narudžbe na API. Nakon što je narudžba uspješno spremljena na serveru, Axios bi vratio odgovor koji bi se zatim koristio za ažuriranje stanja aplikacije.

Korištenje Axiosa u ovoj aplikaciji omogućilo je učinkovito upravljanje zahtjevima i ažuriranjima na strani klijenta. Biblioteka je fleksibilna i može se lako prilagoditi bilo kojem backend API-u. Axioseva jednostavnost i efikasnost u radu s HTTP zahtjevima omogućila je brzo razvijanje web aplikacije i poboljšanje korisničkog iskustva.

```
async getTableNumber(){
  await axios.get('https://toni-web.com/thepurplehat/tables/tableNumber')
  .then((result) => {
    this.tableNumber = result.data
  })
}
```

Slika 4.7. *Primjer Axios asinkrone funkcije za dohvaćanje broja stola.*

4.6. PHP

PHP je serverski skriptni jezik koji se široko koristi za razvoj web aplikacija. Može se koristiti za izradu dinamičkih web stranica i web aplikacija koje su sposobne komunicirati s bazama podataka poput MySQL-a.

Kada se koristi PHP u kombinaciji s MySQL bazom podataka, koristi se za izvršavanje SQL upita i manipuliranje podacima u bazi podataka. On se također koristi za dohvaćanje, dodavanje, ažuriranje i brisanje podataka iz MySQL baze podataka.

PHP pruža mnoge ugrađene funkcije koje olakšavaju rad sa bazom podataka, poput funkcija za stvaranje veze sa istom, izvršavanje upita, dohvaćanje podataka i rad s transakcijama [14].

Za spajanje na MySQL bazu podataka s pomoću PHP-a, koristimo funkciju *mysqli_connect()* (slika 4.8) ili PDO (engl. *PHP Data Objects*) klasu koja omogućuje rad s različitim bazama podataka, uključujući MySQL.

PHP se koristi i za implementaciju autentifikacije i autorizacije korisnika te spremanje i dohvaćanje podataka iz baze podataka koje su potrebne za autentifikaciju. Na primjer, PHP se može koristiti za provjeru korisničkih podataka prilikom prijave u sustav i za spremanje podataka o korisniku nakon prijave.

Ukratko, PHP je serverski skriptni jezik koji se široko koristi u kombinaciji s MySQL bazom podataka za razvoj dinamičkih web aplikacija te omogućuje izvršavanje SQL upita, manipuliranje podacima u bazi podataka i implementaciju autentifikacije i autorizacije korisnika.

```
// Postavke baze podataka
$server = "localhost";
$username = "korisnik";
```

```php
$password = "lozinka";
$dbname = "moja_baza";
// Spajanje na bazu podataka
$conn = mysqli_connect($server, $username, $password, $dbname);
// Provjera povezivanja
if (!$conn) {
    die("Neuspjelo povezivanje na bazu podataka: " . mysqli_connect_error());
}
echo "Uspješno povezivanje na bazu podataka!";
```

Slika 4.8. *Primjer povezivanja na bazu podataka s pomoću mysql_connect funkcije.*

4.7. SQL

SQL (engl. *Structured Query Language*) je jezik za upravljanje relacijskim bazama podataka te je razvijen za manipulaciju i upravljanje podacima koji se pohranjuju u iste.

Nudi mnoge mogućnosti za izvršavanje različitih operacija nad bazom podataka, poput dohvaćanja podataka, izmjene podataka, brisanja podataka i dodavanja novih podataka u bazu. SQL također pruža mogućnost izrade tablica u bazi podataka, definiranje ograničenja i izrada upita za dohvaćanje podataka iz više tablica.

Osnovni elementi SQL-a su tablice, stupci, redci i ključevi. Tablice su temelj baze podataka i sadrže podatke pohranjene u redovima i stupcima. Stupci definiraju vrste podataka koje se pohranjuju u tablici, dok redci predstavljaju pojedinačne zapise u tablici. Ključevi se koriste za povezivanje podataka između različitih tablica.

SQL nudi mnoge mogućnosti za upravljanje bazama podataka, a najčešće se koristi za upravljanje podacima u relacijskim bazama podataka, poput MySQL-a, Oracle-a i Microsoft SQL Servera. SQL se također koristi u kombinaciji s drugim jezicima za razvoj web aplikacija, poput PHP-a i JavaScripta.

Možemo zaključiti kako je SQL standardizirani jezik za upravljanje relacijskim bazama podataka i omogućuje programerima jednostavno i učinkovito upravljanje podacima u bazi podataka [15].

4.7.1. MySQL

MySQL je sustav otvorenog koda za upravljanje relacijskim bazama podataka (engl. *RDBMS*) koji se često koristi za izradu web aplikacija. Ovaj sustav koristi SQL kao jezik za upravljanje bazama podataka.

Može se koristiti na različitim platformama, uključujući Linux, Windows i Mac OS. Nudi mnoge mogućnosti za upravljanje bazom podataka, uključujući podršku za transakcije, referencijalni integritet, sigurnost, indeksiranje podataka i još mnogo toga.

Za upravljanje MySQL-om se obično koriste različiti alati, uključujući MySQL Workbench, phpMyAdmin (slika 4.9) i mnoge druge. Ovi alati omogućuju programerima da stvore i upravljaju tablicama, indeksima, ograničenjima, sigurnošću i drugim aspektima baze podataka.

PHP se često koristi za povezivanje s MySQL bazom podataka i za izvršavanje operacija na istoj te nudi mnoge funkcije njeno upravljanje, uključujući funkcije za povezivanje na nju, izvršavanje upita, čitanje podataka i pisanje podataka.

MySQL se često koristi u kombinaciji s drugim tehnologijama za izradu web aplikacija poput JavaScripta, Vue.js-a, CSS-a, HTML-a i drugih te nudi brzo i učinkovito upravljanje podacima i omogućuje programerima da izrade web aplikacije koje brzo odgovaraju na korisničke zahtjeve i podržavaju veliki broj korisnika [16].

	id	name	price	inStock	description	sold	type	mainType	dependency
Edit Copy Delete	1	Decaffeinated Coffee	2.00	100.00	cup	0.00	hot	drink	NULL
Edit Copy Delete	2	Cocoa	1.20	100.00	cup	1.00	hot	drink	NULL
Edit Copy Delete	3	Hot Chocolate	1.50	100.00	cup	0.00	hot	drink	NULL
Edit Copy Delete	11	Coffee	1.20	100.00	cup	3.00	hot	drink	NULL
Edit Copy Delete	20	Tea Chamomile	1.20	100.00	cup	8.00	hot	drink	NULL
Edit Copy Delete	21	Tea Mint	1.20	100.00	cup	0.00	hot	drink	NULL
Edit Copy Delete	26	Tea Black	1.20	100.00	cup	0.00	hot	drink	NULL
Edit Copy Delete	40	Cappucinno Classic	1.30	100.00	cup	1.00	hot	drink	NULL
Edit Copy Delete	41	Cappucinno Chocolate	1.30	100.00	cup	1.00	hot	drink	NULL
Edit Copy Delete	42	Cappucinno Vanilla	1.30	100.00	cup	1.00	hot	drink	NULL
Edit Copy Delete	45	Cappucinno Irish	1.30	100.00	cup	5.00	hot	drink	NULL
Edit Copy Delete	51	Coca-Cola	1.90	100.00	0.33l	2.00	juice	drink	NULL
Edit Copy Delete	52	Coca-Cola Zero	1.90	100.00	0.33l	0.00	juice	drink	NULL
Edit Copy Delete	53	Fanta	1.90	100.00	0.33l	0.00	juice	drink	NULL
Edit Copy Delete	54	Sprite	1.90	100.00	0.33l	1.50	juice	drink	NULL

Slika 4.9. *Prvih nekoliko artikala u bazi podataka.*

4.8. Visual Studio Code

Visual Studio Code je besplatni program za uređivanje koda koji se može koristiti na različitim operativnim sustavima, uključujući Windows, Linux i Mac OS. Nudi mnoge značajke za programere, uključujući brzo kodiranje, inteligentni kod, ispravke tijekom pisanja, duboko pretraživanje i refaktoriranje.

Jedna od najvažnijih značajki Visual Studio Code-a je sposobnost dodavanja proširenja i dodataka kako bi se poboljšala i prilagodila njegova funkcionalnost. Korisnici mogu preuzeti i instalirati različita proširenja koje omogućuju razne značajke, uključujući upravljanje Gitom, JavaScript, Vue.js, CSS, HTML, PHP i mnoge druge.

Uz Visual Studio Code, programeri mogu brzo i jednostavno izraditi i uređivati kod te pružiti detaljno debagiranje i ispravke tijekom pisanja. Ovaj program ima bogat sustav tema i mogućnost prilagodbe korisničkog sučelja kako bi odgovarao potrebama korisnika.

Omogućuje stvaranje i upravljanje različitim projektima u okviru jednog programa te dijeljenje koda i projekata s drugima. Sve ove značajke čine Visual Studio Code popularnim alatom za programiranje i uređivanje koda, a ponajviše za izradu web aplikacija [17].

5. IZRADA APLIKACIJE

U ovom radu je korištena kombinacija backend i frontend tehnologija kako bi se izradila web aplikacija za narudžbu hrane u restoranu putem QR kodova [18].

Nakon toga napravljen je backend za prijavu korisnika, u kojem je uključena i logika s bazom podataka. Nakon što se korisnik prijavi, ovisno o njegovoj ulozi (korisnik, konobar, kuhar, administrator ili generator) postoji određena logika za autorizaciju.

Za korisnika je napravljen frontend s cjenikom za naručivanje hrane, košaricom za narudžbe i funkcionalnošću za naručivanje hrane. Backend odrađuje logiku i povezan je s bazom podataka za naručivanje hrane.

Za konobara je izrađen frontend koji mu omogućuje pregledavanje narudžbi, označavanje naručene hrane kao pripremljene i slanje obavijesti korisniku da je hrana spremna. Tada je na backend dodano povezivanje s bazom podataka za spremanje narudžbi i ažuriranje statusa narudžbe.

Za kuhara je izrađen frontend koji mu omogućuje pregledavanje pripremljenih narudžbi i označavanje naručene hrane kao gotove te je na jednak način kao kod konobara odrađeno povezivanje s bazom podataka za prikazivanje pripremljenih narudžbi i ažuriranje statusa narudžbe.

Dodana je nova vrsta prijave naziva "generator". Za njega je izrađen frontend i backend koji se koristi za generiranje slučajnog četveroznamenkastog koda koji se mijenja svakih 30 sekundi. Taj kod se prikazuje na ekranu u restoranu, a korisnik ga mora upisati prilikom potvrđivanja narudžbe kako bi potvrdio da se nalazi u restoranu.

Prijava korisnika ide putem QR kodova, tako da korisnici mogu jednostavno skenirati kod za svojim stolom kako bi naručili hranu. Za to je na frontend dodan Vue 3 paket za otvaranje kamere te je prilagođen backend.

Nakon izrade frontend-a i logike za sve uloge, kreiran je frontend za administratora. Administrator ima pristup svim funkcijama, može pregledavati popis narudžbi, uređivati cjenik, može dodavati nove korisnike, konobare i kuhare i dodavati nove artikle u cjenik.

Zatim je u PHP-u kreiran backend funkcionalnost za administriranje, uključujući i pristup bazi podataka za pregledavanje, upravljanje narudžbama i artiklima u cjeniku te dodavanje korisnika, konobara i kuhara.

Kako bi sve uloge u sustavu mogle pravilno funkcionirati, međusobno su povezane prilikom narudžbi. Konobar i kuhar dobivaju obavijesti o novim narudžbama, a administrator može pregledati sve narudžbe i upravljati artiklima u cjeniku. Generator se koristi za potvrdu nalazili se korisnik stvarno u restoranu i sprečavanje zlouporabe QR kodova.

Nakon završetka izrade, veliki dio posvećen je ispravljanju grešaka te uklanjanju istih i upozorenja. Aplikacija je testirana na različitim uređajima i preglednicima te je zbrinuto da se sve funkcionalnosti i elementi prikazuju ispravno i da su dostupni svim korisnicima.

Ukupno gledajući, ovaj projekt obuhvaćao je izradu frontenda i backenda te povezivanje baze podataka kako bi se omogućila funkcionalnost narudžbi u restoranu. Implementirano je više uloga kako bi se svaka osoba u restoranu mogla koristiti sustavom na način koji odgovara njezinoj ulozi, te je pobrinuto da sve funkcionalnosti budu dostupne korisnicima kao i administratoru koji upravlja sustavom.

5.1. Baza podataka

MySQL je besplatan otvoreni programski relacijski sustav za upravljanje bazama podataka, koji je korišten za ovaj projekt zajedno s PHP-om kao backendom. Baza podataka je osmišljena kako bi olakšala spremanje i upravljanje podacima vezanim uz artikle, korisnike, narudžbe, konobare, kuhare i administratore. Korišten je MySQL kao osnova za jednostavno spajanje baze podataka s aplikacijom, kako bi se učinkovito pohranili podaci i pristupilo im se kad god su potrebni.

Lozinke korisnika pohranjuju se u tablici ‘Users’ bez enkripcije hash algoritmom. Međutim, korištena je MD5 enkripcija za spremanje lozinke u bazi podataka kako bi se osiguralo da lozinke neće biti vidljive u slučaju da netko izvan sustava dobije pristup bazi podataka. To je prilično osnovna razina zaštite, a mogu se primijeniti i druge napredne tehnike enkripcije kako bi se dodatno poboljšala

sigurnost podataka. U budućnosti bi bilo dobro ažurirati aplikaciju kako bi se koristila jača metoda enkripcije poput bcrypt ili Argon2a.

Strukture dviju glavnih MySQL tablica koje su korištene za ovaj projekt, ‘Users’ predstavlja sve vrste korisnika s njihovim podacima (SQL struktura vidljiva na slici 5.1), a ‘Products’ sa artiklima, odnosno hranom i pićem s kojima restoran raspolaže (SQL struktura vidljiva na slici 5.2). Više o samoj inicijalizaciji i strukturi varijabli, njihovim primarnim i unikatnim ključevima te početnim vrijednostima može se vidjeti na slikama 5.1 i 5.2.

"Users":

- id: jedinstveni identifikator korisnika
- firstname: ime korisnika
- lastname: prezime korisnika
- fullname: puno ime korisnika
- username: korisničko ime korisnika
- email: jedinstvena e-mail adresa korisnika
- password: lozinka korisnika (MD5 enkriptirana)
- role: razina pristupa korisnika
- verified: status verifikacije korisnika
- verificationLink: verifikacijska poveznica
- token: sigurnosni token korisnika

"Products":

- id: jedinstveni identifikator proizvoda
- name: ime proizvoda
- price: cijena proizvoda
- inStock: količina proizvoda koja je trenutno dostupna
- description: kratki opis proizvoda
- sold: količina proizvoda koja je prodana
- type: tip proizvoda (npr: sok, pivo, vino…)
- mainType: glavni tip proizvoda (piće i hrana)
- dependency: ovisnost proizvoda o drugom proizvodu putem id-a

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	firstname	varchar(30)	utf8_unicode_ci		Yes	NULL			Change Drop More
☐	3	lastname	varchar(30)	utf8_unicode_ci		Yes	NULL			Change Drop More
☐	4	fullname	varchar(100)	utf8_unicode_ci		No	None			Change Drop More
☐	5	username	varchar(100)	utf8_general_ci		Yes	NULL			Change Drop More
☐	6	email	varchar(100)	utf8_unicode_ci		Yes	NULL			Change Drop More
☐	7	password	varchar(100)	utf8_unicode_ci		No	None			Change Drop More
☐	8	role	varchar(20)	utf8_general_ci		Yes	NULL			Change Drop More
☐	9	verified	int(11)			No	0			Change Drop More
☐	10	verificationLink	varchar(100)	utf8_unicode_ci		Yes	NULL			Change Drop More
☐	11	token	varchar(100)	utf8_unicode_ci		Yes	NULL			Change Drop More

Slika 5.1 *Struktura 'Users' tablice u bazi podataka.*

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	id	int(6)			No	None		AUTO_INCREMENT	Change Drop More
☐	2	name	varchar(75)	utf8mb4_general_ci		No	None			Change Drop More
☐	3	price	decimal(8,2)			No	None			Change Drop More
☐	4	inStock	decimal(8,2)			No	100.00			Change Drop More
☐	5	description	varchar(200)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	6	sold	decimal(8,2)			No	0.00			Change Drop More
☐	7	type	varchar(30)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	8	mainType	varchar(30)	utf8mb4_general_ci		Yes	NULL			Change Drop More
☐	9	dependency	int(6)			Yes	NULL			Change Drop More

Slika 5.2 *Struktura 'Products' tablice u bazi podataka.*

Određena pravila kod 'User' tablice podataka:

- firstname - koristi se samo kod administratora, konobara i kuhara
- lastname - koristi se samo kod administratora, konobara i kuhara
- fullname - kod administratora, konobara i kuhara je skup firstname-a i lastname-a, dok kod gostiju predstavlja broj stola (primjer: 'Table 2')
- email - obavezan samo kod administratora, kod drugih je proizvoljan
- verified - iako je tip varijable int(11), kasnije je preuređen da radi kao boolean te se u njega zapisuju stanja 0 - neverificiran i 1 - verificiran

Određena pravila kod ‘Products’ tablice podataka:

- firstname - koristi se samo kod administratora, konobara i kuhara
- lastname - koristi se samo kod administratora, konobara i kuhara
- fullname - kod administratora, konobara i kuhara je skup firstname-a i lastname-a, dok kod gostiju predstavlja broj stola (primjer: ‘Table 2’)
- email - obavezan samo kod administratora, kod drugih je proizvoljan
- verified - iako je tip varijable int(11), kasnije je preuređen da radi kao boolean te se u njega zapisuju stanja 0 - neverificiran i 1 - verificiran
- verificationLink - koristio se samo u prvoj fazi izrade projekta dok nije bilo prijave putem QR koda prilikom resetiranja lozinke
- token - koristio se samo u prvoj fazi izrade projekta dok nije bilo prijave putem QR koda prilikom same prijave (tu se spremao token za provjeru)

5.2. Spremanje narudžbi - REST API

Dio web aplikacije koji se odnosi na spremanje narudžbi napravljen je kao dio REST API-ja u JSON (engl. *JavaScript Object Notation*) formatu (slika 5.3). REST (engl. *Representational State Transfer*) je arhitektura koja opisuje način na koji se podaci mogu prenositi između servera i klijenta preko HTTP protokola. REST API se sastoji od HTTP funkcija (engl. *GET, POST, PUT, DELETE*) koji predstavljaju CRUD (engl. *Create, Read, Update, Delete*) operacije nad podacima, a korisnik pristupa API-u putem linka.

U ovom slučaju, REST API je omogućio korisnicima da putem mobilne aplikacije naručuju hranu i piće u restoranu. Za implementaciju REST API-ja korišten je PHP programski jezik. Svakoj funkcionalnosti u aplikaciji pridružena je HTTP metoda koja se poziva iz JavaScript koda u mobilnoj aplikaciji putem Axiosa. Na primjer, za spremanje narudžbe koristi se POST metoda.

Kod za svaku funkcionalnost u REST API-ju sastoji se od nekoliko dijelova. Prvo se provjerava autorizacija korisnika kako bi se spriječio pristup nepoželjnim korisnicima. Nakon toga slijedi provjera valjanosti unesenih podataka te ako su podaci ispravni, PHP skripta vrši spremanje

podataka u predviđeni file u JSON formatu (slika 5.3). U slučaju da se dogodi greška, PHP skripta vraća odgovarajući HTTP status kod, koji se zatim obrađuje u JavaScript kodu.

Kao dio REST API-ja, stvorena je funkcionalnost za spremanje narudžbi, koja se koristi za pohranjivanje detalja o narudžbama u bazu podataka. Svaka narudžba povezana je s određenim korisnikom, stolom i naručenim artiklima.


```
[
  "[{\"waiter\":\"true\",\"tableName\":\"table1\",\"payment\":\"true\",\"id\":\"141\",
  \"name\":\"Jack Daniels 0.03l\",\"price\":\"15.00\",\"quantity\":\"2\",\"sequence\":
  \"1\", \"time\":\"1673564309\",\"type\":\"drink\"}]"
]
```

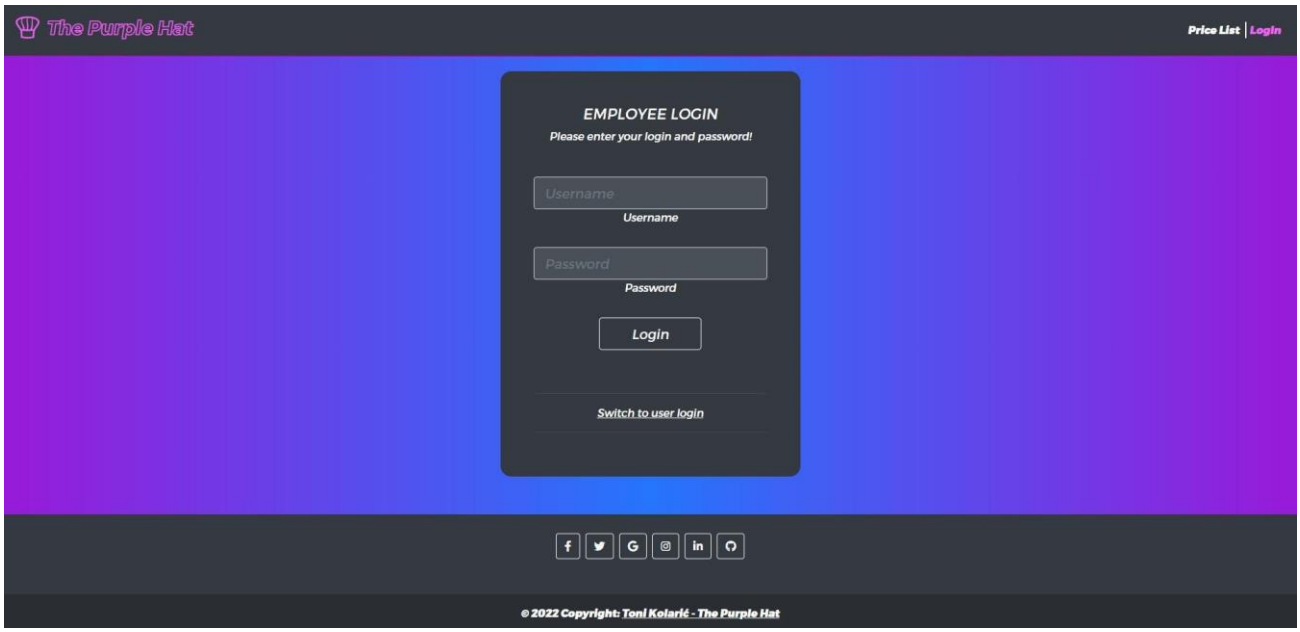

Slika 5.3 *Prikaz jedne narudžbe u JSON formatu.*

5.3. Prijava u aplikaciju

Aplikacija za narudžbu hrane i pića u restoranu omogućuje prijavu različitim vrstama korisnika putem stranice za prijavu. Gosti restorana, koji ne posjeduju korisnički račun, mogu se prijaviti putem QR koda koji se nalazi na stolu. Nakon što gost skenira QR kod kamerom svog mobilnog uređaja, automatski će biti preusmjeren na stranicu za narudžbu hrane i pića, gdje će mu biti omogućeno naručivanje artikala.

Korisnici koji posjeduju korisnički račun u aplikaciji, poput konobara, kuhara, administratora i generatora, mogu se prijaviti putem stranice za prijavu. Stranica za prijavu nalazi se na početnoj stranici aplikacije (slika 5.4) i omogućuje korisnicima da unesu svoje korisničko ime i lozinku kako bi pristupili svojim računima. Nakon uspješne prijave, korisniku će biti omogućen pristup svojim funkcionalnostima u aplikaciji.

Korisnike je moguće kreirati samo s administratorskog računa. Isti ima mogućnosti brisanja već postojećih računa. Na taj način samo on može upravljati brojem stolova i zaposlenika u restoranu što je svakako najbolja opcija za ovu aplikaciju.



Slika 5.4 *Prijava u aplikaciju.*

5.4. Stvaranje i brisanje računa

Stvaranje i brisanje korisničkih računa u aplikaciji može obaviti samo administrator koji ima ovlasti za upravljanje korisnicima. Kada je administrator prijavljen u aplikaciju, može pristupiti stranici za upravljanje korisnicima. Na toj stranici administrator ima mogućnost dodavanja i brisanja postojećih korisničkih računa.

5.4.1. Stvaranje računa

Kada administrator odluči dodati novi korisnički račun, on mora ispuniti obrazac s osnovnim podacima o korisniku. Potrebno je provjeriti ispravnost svih unesenih podataka prije nego što se novi korisnički račun stvori u bazi podataka.

Stvaranje novog korisničkog stola u restoranu: odvija se na administratorskoj stranici ‘Tables’ jednostavnim pritiskom na tipku ‘Add’ nakon čega se potvrđuje željena akcija ponovnim pritiskom na tipku ‘Add’ iz novonastalog okvira (trenutno je administratoru omogućeno dodavanje maksimalno 10 stolova).

Stvaranje novog računa za zaposlenika u restoranu: odvija se na administratorskoj stranici ‘Employees’ jednostavnim pritiskom na tipku ‘Add new’ kod željene uloge (konobar ili kuhar) nakon čega se otvara novi okvir, odnosno forma za popunjavanje podataka: firstname, lastname, username, password i repeat.

Generatore nije u mogućnosti kreirati jer su automatski kreirani prilikom izrade aplikacije.

5.4.2. Brisanje računa

Kada se administrator odluči obrisati postojeći korisnički račun, mora se nalaziti na svojim administratorskim stranicama, te je u mogućnosti obrisati sve uloge osim generatora koji su automatski generirani prilikom nastanka aplikacije.

Brisanje postojećeg korisničkog stola u restoranu: odvija se na administratorskoj stranici ‘Tables’ jednostavnih pritiskom na tipku ‘Remove’ nakon čega se potvrđuje željena akcija ponovnim pritiskom na tipku ‘Delete’ iz novonastalog okvira.

Brisanje postojećeg računa za zaposlenika u restoranu: odvija se na administratorskoj stranici ‘Employees’ jednostavnih pritiskom na tipku ‘Delete’ kod željenog konobara ili kuhara (lista konobara se nalazi lijevo, a kuhara desno).

5.5. Stvaranje, brisanje i ažuriranje artikla

Stvaranje, ažuriranje i brisanje artikala u aplikaciji može se obaviti samo od strane administratora, koji ima ovlasti za upravljanje artiklima. Kada je administrator prijavljen u aplikaciju, može pristupiti stranici za upravljanje artiklima. Na toj stranici administrator ima mogućnost dodavanja, brisanja i uređivanja postojećih artikala.

5.5.1. Stvaranje artikla

Kada administrator odluči dodati novi artikli, on mora ispuniti obrazac s osnovnim podacima o njemu. Potrebno je provjeriti ispravnost svih unesenih podataka prije nego što se novi artikl stvori u bazi podataka.

Stvaranje novog artikla u restoranu se odvija na administratorskoj stranici ‘Products’ (slika 5.5) jednostavnih pritiskom na tipku ‘Add’ nakon čega se iz padajućeg izbornika odabire glavni tip artikla (piće ili hrana).

Ako je odabrano dodavanje pića: otvara se forma za popunjavanje: name, price (valuta je euro €), amount i type. ‘Name’ i ‘price’ se unose tipkovnicom (‘name’ - svi znakovi, ‘price’ - decimalni ili obični broj), a ‘amount’ i ‘type’ su padajući izbornici:

- amount - označava prodajnu količinu pića, te se može birati između: šalice, čaše, 0.03l, 0.2l, 0.33l, 0.5l i 1.0l
- type - predstavlja tip pića: topli, sok, pivo, alkohol (predstavlja žestice) i vino

Ako je odabrano dodavanje hrane: otvara se forma za popunjavanje: name, price (valuta je euro €), description i type. ‘Name’, ‘price’ i ‘description’ se unose tipkovnicom (‘name’ - svi znakovi, ‘price’ - decimalni ili obični broj, ‘description’ - opis/sastojci) dok je type padajući izbornik:

- type - predstavlja tip hrane: juha, pizza, grill, prilog i desert

Slika 5.5 *Stvaranje novog artikla na administratorskim stranicama.*

5.5.2. Ažuriranje artikla

Kada administrator odluči ažurirati artikli, on mora odabrati jedan od postojećih kojega želi ažurirati, odnosno izmijeniti. Prilikom ažuriranja svim unesenim podacima se provjerava valjanost prije nego što se novi artikl doda u bazu podataka.

Ažuriranje postojećeg artikla (slika 5.6) se odvija na administratorskoj stranici 'Products' jednostavnih pritiskom na tipku 'Edit' nakon čega se iz padajućeg izbornika odabire glavni tip artikla (piće ili hrana), a zatim jedan od ponuđenih, odnosno postojećih artikala odabranog tipa. Nakon toga se otvara forma sa podacima artikala koje možemo izmijeniti. Spremanje željenih izmjena potvrđujemo pritiskom na tipku 'Edit'.

Prilikom odabira pića i hrane razlikuju se forme za ispunjavanje podataka kao i prilikom stvaranja istih.

Ako je odabrano piće: otvara se forma za popunjavanje: name, price (€), amount i type. 'Name' i 'price' se unose tipkovnicom ('name' - svi znakovi, 'price' - decimalni ili obični broj), a 'amount' i 'type' su padajući izbornici:

- amount - označava prodajnu količinu pića, te se može birati između: šalice, čaše, 0.03l, 0.2l, 0.33l, 0.5l i 1.0l
- type - predstavlja tip pića: topli, sok, pivo, alkohol (predstavlja žestice) i vino

Ako je odabrana hrana: otvara se forma za popunjavanje: name, price (€), description i type. 'Name', 'price' i 'description' se unose tipkovnicom ('name' - svi znakovi, 'price' - decimalni ili obični broj, 'description' - opis/sastojci) dok je type padajući izbornik:

- type - predstavlja tip hrane: juha, pizza, grill, prilog i desert

Slika 5.6 *Ažuriranje postojećeg artikla na administratorskim stranicama.*

5.5.3. Brisanje artikla

Kada se administrator odluči obrisati postojeći artikl, mora se nalaziti na svojim administratorskim stranicama.

Brisanje postojećeg artikla odvija se na administratorskoj stranici ‘Products’ jednostavnih pritiskom na tipku ‘Delete’ nakon čega se iz padajućeg izbornika odabire glavni tip artikla (piće ili hrana), a zatim jedan od ponuđenih, odnosno postojećih artikala odabranog tipa. Nakon toga se potvrđuje željena akcija ponovnim pritiskom na tipku ‘Delete’ iz novonastalog okvira.

5.6. Statistički prikaz artikala

Prikaz narudžbi jedna je od ključnih funkcionalnosti u aplikaciji za narudžbu hrane i pića u restoranu. Mogućnost pregleda i praćenja narudžbi omogućava konobarima, kuharima i administratorima da učinkovitije organiziraju proces naručivanja i posluživanja hrane i pića u restoranu. Ova funkcija posebno je važna za administratore koji mogu pregledavati sve narudžbe iz svih stolova u restoranu.

Narudžbe se prikazuju na dva načina: "Finished orders" i "Active orders". Prikaz narudžbi raspoređen je od najnovijeg prema najstarijem, kako bi se osiguralo da su najaktualnije narudžbe vidljive na vrhu popisa. Ova funkcionalnost olakšava konobarima i kuharima da budu upoznati s trenutnim narudžbama i njihovim prioritetima.

Odabir željene opcije iz padajućeg izbornika kod administratora omogućava prikaz određene vrste narudžbi, a to može biti "Finished orders" ili "Active orders". Ovisno o odabranoj opciji, prikazuju se sve narudžbe koje su prikupljene u bazi podataka odabranog tipa. Ova funkcionalnost pomaže u učinkovitom upravljanju narudžbama, smanjenju grešaka i povećanju kvalitete usluge u restoranu.

Konobari i kuhari u aplikaciji vide samo narudžbe koje su im dodijeljene ili koje su još uvijek otvorene. To znači da vide samo one narudžbe koje još nisu označene kao poslužene ili pripremljene. Ovo omogućuje da konobari i kuhari imaju uvid u samo one narudžbe koje su im relevantne te da se ne zamaraju nepotrebnim informacijama koje im ne bi koristile u obavljanju njihovih zadataka. Kada konobar ili kuhar označi narudžbu kao pripremljenu ili posluženu, ta narudžba se automatski prebacuje u kategoriju 'Finished orders' na administratorskom sučelju. Ova funkcionalnost omogućuje

konobarima i kuharima jednostavniji i pregledniji prikaz narudžbi te olakšava njihov rad i donošenje odluka prilikom pripreme hrane i pića.

5.7. Generator slučajnih četveroznamenkastih brojeva

Generator se koristi za generiranje četveroznamenkastih brojeva koji se koriste za potvrdu narudžbe korisnika. Generiranje novog broja se obavlja svakih 30 sekundi, a stari broj se još 30 sekundi čuva u bazi podataka u slučaju da korisnik kasnije želi unijeti broj ili dođe do nekog problema s povezivanjem. Generator je povezan s MySQL bazom podataka za spremanje generiranih brojeva.

Na početku, kada se aplikacija pokrene, generator se prijavi na stranicu za prijavu upisivanjem svojih vjerodajnica. Nakon toga, na velikom zaslonu u restoranu prikazuje se trenutni broj koji korisnici moraju unijeti kako bi potvrdili svoju narudžbu.

Generatorova logika je izrađen korištenjem PHP-a, a napisan je i prilagođen za rad s Vue.js sučeljem koji se koristi za frontend. Uz redovito generiranje novih brojeva, generator također provjerava povezivost s bazom podataka i u slučaju problema s povezivanjem obavještava korisnike da pokušaju ponovno kasnije.

Uz pomoć ovog generatora, korisnici restorana mogu jednostavno i brzo potvrditi svoju narudžbu, što poboljšava njihovo iskustvo u restoranu i smanjuje vrijeme čekanja na uslugu.

5.8. Logika naručivanja

Logika naručivanja u restoranu uključuje tri glavna aktera: gosta, konobara i kuhara. Gosti obično sjede za stolom i pregledavaju jelovnik, a zatim naručuju hranu i piće od konobara. Konobari tada prenose narudžbu kuharu, koji priprema hranu i piće. Nakon pripreme, konobar dostavlja naručeno jelo i piće gostima.

U aplikaciji za naručivanje hrane i pića u restoranu, gosti mogu koristiti svoj mobilni uređaj kako bi izravno pregledali jelovnik i naručili hranu i piće. Narudžba se tada automatski prikazuje na zaslonu za konobare i kuhare koji mogu pregledati nedovršene narudžbe i pripremiti jela i piće. Nakon pripreme, konobari mogu označiti narudžbu kao posluženu, a kuhari kao napravljenu.

Kada gost skenira QR kod i napravi narudžbu, postoji mogućnost da želi dodati još neke stavke nakon prvotne narudžbe. U tom slučaju, gost može ponovno skenirati isti QR kod i dodati nove stavke

u svoju narudžbu. Nakon toga, nova narudžba se automatski prikazuje na zaslonu konobara i kuhara, kako bi znali da ima nove stavke koje trebaju pripremiti. Ovo je praktično rješenje koje omogućuje gostima da u svoju narudžbu dodaju nove stavke bez potrebe za interakcijom s konobarom.

Gosti također skeniraju QR kod na kraju kako bi zatražili račun. Kada gost skenira QR kod, putem aplikacije zatraži plaćanje. Tada gost može pregledati sve stavke koje je naručio i provjeriti ukupnu cijenu. Nakon toga, konobar dobiva obavijest da je gost zatražio plaćanje te odlazi do njega kako bi izvršili isto. Ovo je brz i učinkovit način plaćanja, koji olakšava završetak obroka i izlazak iz restorana.

Ova aplikacija za naručivanje hrane i pića u restoranu poboljšava učinkovitost procesa naručivanja i smanjuje vrijeme čekanja gostiju. Također smanjuje potrebu za dodatnim osobljem, jer gosti mogu naručiti hranu i piće sami putem mobilnog uređaja, a konobari i kuhari mogu učinkovitije upravljati narudžbama i pripremiti jela i piće u skladu s njihovim potrebama.

5.9. Generiranje QR kodova

QR kodovi su kvadratni matični bar kodovi koji mogu pohraniti različite vrste informacija, poput URL adresa, teksta, kontaktnih podataka ili čak multimedijskih datoteka. Na stranici koju proučavamo. QR kodovi generiraju se pomoću naprednih algoritama i alata.

QR kodovi su izgenerirani putem vanjske web aplikacije tako da u sebi nose podatke broja stola i lozinku za njegovu prijavu kako bi prošli autentifikaciju i autorizaciju. Nakon uspješno odrađene autentifikacije i autorizacije korisnik pristupa stranicama za narudžbu.

6. KORIŠTENJE APLIKACIJE

Za korisničko korištenje aplikacije potrebno je mobilnim uređajem skenirati QR kod koji se nalazi na stolu u restoranu. Nakon toga, korisnik može dodati željene stavke u košaricu, a zatim potvrditi narudžbu upisivanjem broja koji se generira na ekranu u restoranu. Ukoliko korisnik želi naručiti nešto dodatno, ponovno skenira QR kod sa stola i ponavlja postupak. Nakon završetka narudžbe, korisnik može zatražiti plaćanje skeniranjem QR koda sa stola te odabirom opcije plaćanja u košarici.

Prilikom prijave konobara, kuhara, administratora i generatora nije moguća prijava putem QR koda, već se odrađuje prijavom kroz stranicu za prijavu na web aplikaciji nakon čega se odrađuje autentifikacija i autorizacija te preusmjeravanje na za to predviđene stranice.

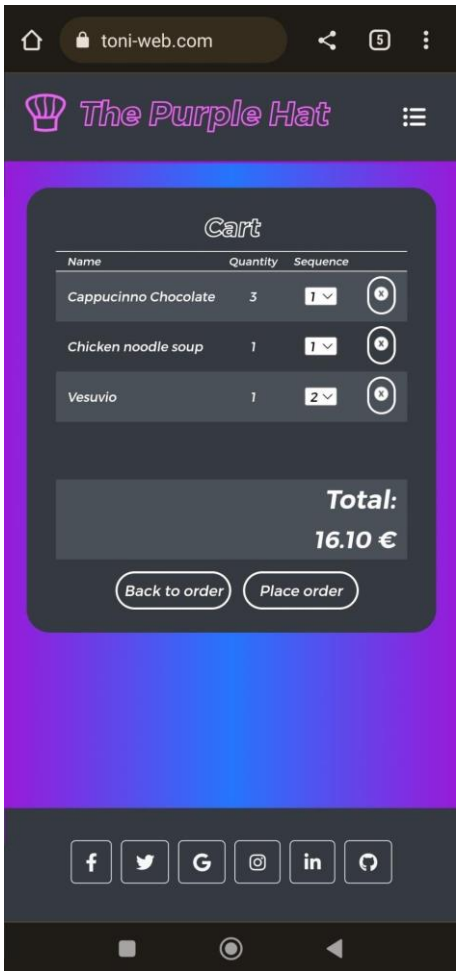
Razvijena web aplikacija je dostupna putem web preglednika poput Google Chrome i Mozilla Firefox te omogućuje korisnicima jednostavan pristup različitim funkcionalnostima putem mreže. Za pravilan rad aplikacije koristi se web poslužitelj za upravljanje zahtjevima od klijenta te aplikacija za izvršavanje zatraženih zadataka.

- Korisnik skenira QR kod sa stola u restoranu kako bi se prijavio.
- Korisnik dodaje hranu, piće te njihovu količinu u košaricu.
- Korisnik prilikom potvrđivanja narudžbe mora upisati četveroznamenkasti slučajno generirani broj sa zaslona u restoranu kako bi potvrdio svoju prisutnost.
- Konobar prima narudžbu pića koju je dužan dostaviti, a kuhinja hranu.
- Ukoliko korisnik želi dodati nove artikle u narudžbu ili zatražiti plaćanje, to mu je omogućeno ponovnim skeniranjem QR koda.
- Konobari, kuhari i administrator se prijavljuju putem korisničkog imena i lozinke.

6.1. Korisnik - gost u restoranu

Proces naručivanja hrane i pića putem aplikacije počinje skeniranjem QR koda sa stola koristeći mobilni uređaj. Nakon što pristupite digitalnom jelovniku, možete dodati željene stavke u košaricu. Potvrda narudžbe vrši se upisivanjem broja koji se generira na ekranu unutar restorana. Ako

želite naručiti dodatne stavke, možete ponovno skenirati QR kod sa stola i ponoviti postupak narudžbe. Kada ste spremni za plaćanje, možete skenirati QR kod sa stola i odabrati opciju 'Zatraži plaćanje' u košarici (slika 6.1). Na kraju je potrebno pričekati konobara koji će doći do vašeg stola kako biste dogovorili način plaćanja.

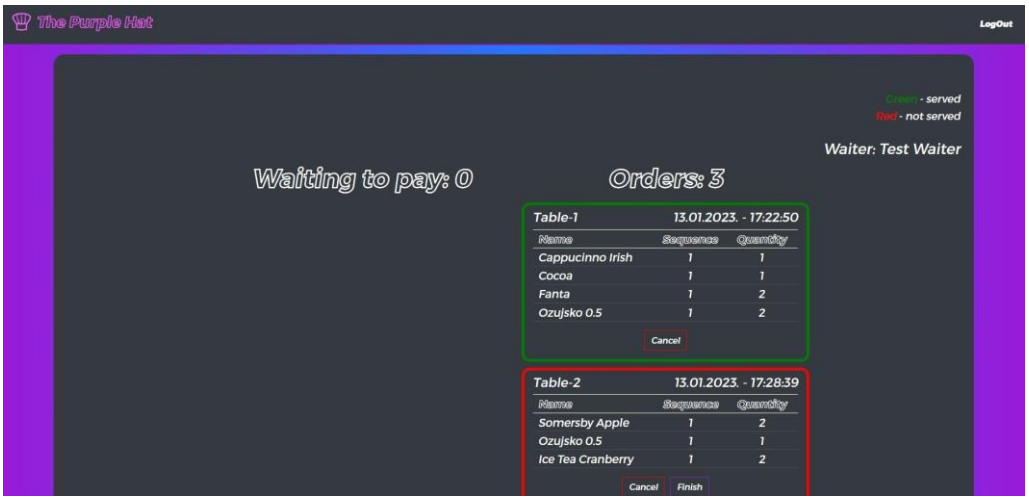


Slika 6.1 *Košarica.*

6.2. Konobar

Konobar se prvo prijavljuje na stranici za prijavu aplikacije. Nakon prijave, primaju obavijesti o novim narudžbama, koje uključuju informacije o stolu i narudžbi pića (slika 6.2). Kada

dobije obavijest, konobar zatim priprema narudžbu kako je navedeno. Nakon pripreme, narudžba se poslužuje na odgovarajući stol. Konobar zatim označava narudžbu kao posluženu u aplikaciji, čime se ažurira status narudžbe. Na kraju smjene isti se odjavljuje iz aplikacije kako bi označio kraj radnog vremena.



Slika 6.2 *Prikaz konobarske stranice.*

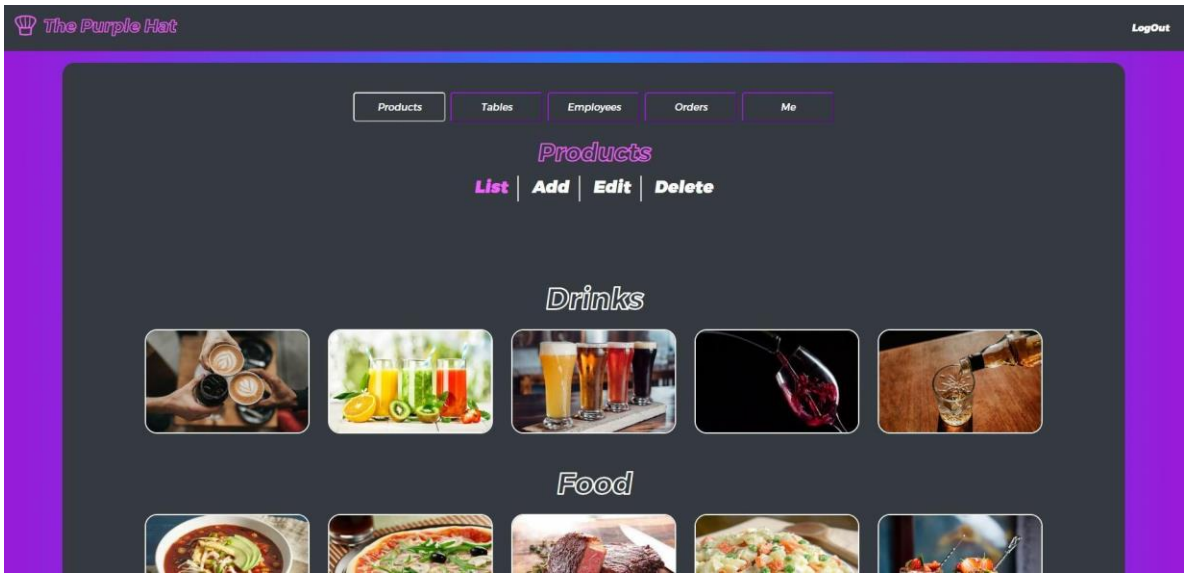
6.3. Kuhar

Proces za kuhara započinje prijavom na aplikaciju. Nakon uspješne prijave, kuha prima obavijesti o novim narudžbama, uključujući informacije o stolu i narudžbi hrane. Slijedi priprema narudžbe prema navedenim specifikacijama. Kada je narudžba spremna, kuhar poziva konobara da narudžbu odnese na odgovarajući stol. Nakon posluživanja, narudžba se u aplikaciji označava kao poslužena, čime se ažurira njen status. Prilikom završetka smjene, kuhar se odjavljuje iz aplikacije, čime označava kraj radnog vremena.

6.4. Administrator

Administrator se prijavljuje na stranicu. Nakon uspješne prijave, pristupa administratorskim stranicama (slika 6.3) te mu se pružaju mogućnosti dodavanja, mijenjanja ili brisanja konobara i

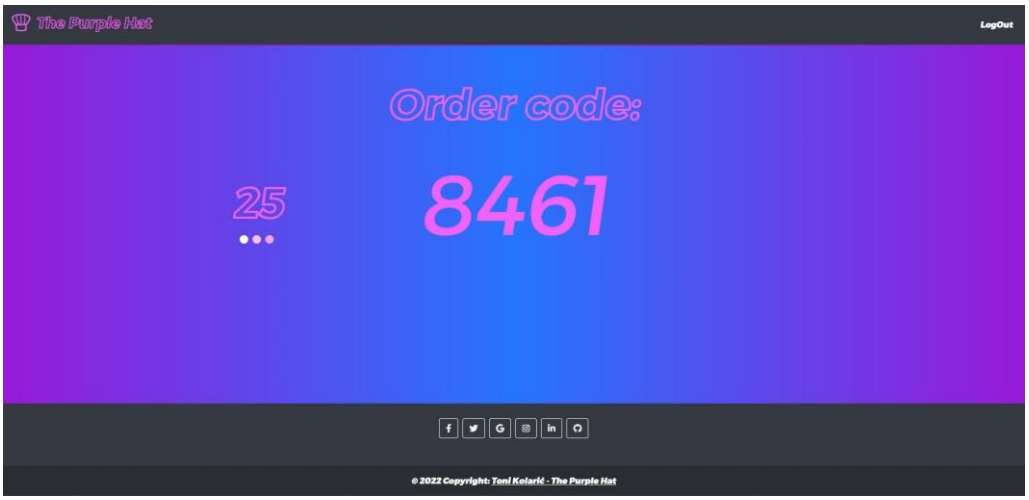
kuhara što pruža fleksibilnost u upravljanju osobljem. Također, mogu dodavati, mijenjati ili brisati artikle, njihova stanja i kategorije, omogućujući učinkovito upravljanje jelovnikom i zalihama. Administrator također ima pristup osnovnoj statistici narudžbi s osnovnim filterima, što pruža uvid u performanse restorana. Kada završi s radom, administrator se odjavljuje iz aplikacije.



Slika 6.3 *Administratorske stranice.*

6.5. Generator

Generator se prijavljuje na stranicu za prijavu na velikom zaslonu u restoranu koji je predviđen za tu svrhu (slika 6.4). Nakon uspješne prijave, proces automatskog generiranja novog četveroznamenkastog broja odvija se svakih 30 sekundi. Svaki novi broj se automatski sprema u bazu podataka koja je povezana na MySQL. U isto vrijeme, aplikacija čuva prethodno izgenerirani broj idućih 30 sekundi, kako bi se mogli riješiti potencijalni problemi ili kašnjenja korisnika pri upisu. Na kraju radnog dana, generator se odjavljuje iz aplikacije, čime se označava kraj njegovog radnog vremena.



Slika 6.4 *Prikaz generatora.*

7. ZAKLJUČAK

U ovom završnom radu predstavljena je izrada web aplikacije za restoran koja omogućuje pojednostavljeno i ubrzano naručivanje hrane i pića unutar istog te njihovo pripremanje i posluživanje. Za razvoj aplikacije korištene su frontend tehnologije kao što su HTML, CSS i JavaScript, a za backend je korištena PHP tehnologija za povezivanje na MySQL bazu podataka. Za implementaciju funkcionalnosti korištena je Vue tehnologija koja je omogućila izradu dinamičkog i korisnički prijateljskog sučelja.

Jedna od glavnih prednosti ove web aplikacije je omogućavanje korisnicima naručivanje hrane i pića bez kontakta s konobarom te čini proces narudžbe i posluživanja bržim i učinkovitijim. Također, aplikacija omogućuje kuharu i konobaru brzo i jednostavno primanje narudžbi te njihovo pripremanje i posluživanje.

Menufy, GrubHub i Waiterio su slične aplikacije koje korisnicima omogućuju naručivanje hrane putem QR koda. Menufy ima manji izbor restorana, ali nudi brzo naručivanje u samom restoranu. GrubHub ima širi izbor restorana i hrane, ali ne nudi uvijek jednostavno i brzo naručivanje u samom restoranu te može biti potrebno čekanje na dostavu. GrubHub je dostupan u većem broju gradova nego Menufy, ali ovisi o lokalnoj dostupnosti restorana. Trenutno najsličnija ovoj aplikaciji je Waiterio koja pruža gotovo iste usluge.

U usporedbi s ovim aplikacijama, razvijena aplikacija za narudžbu unutar restorana putem QR koda nudi jednostavnost korištenja i brzu narudžbu u samom restoranu. Također, omogućuje korisnicima pristup svojoj košarici, narudžbi novih stavki kao i mogućnosti završavanja narudžbe u svakome trenutku. Nedostatak može biti potreba za određenom infrastrukturom u restoranu kako bi se QR kodovi mogli koristiti, što neke restorane može odvratiti od implementacije ove tehnologije.

Implementirana je i dodatna funkcionalnost za generiranje jedinstvenog četveroznamenkastog koda koji korisnik mora upisati prilikom potvrđivanja narudžbe kako bi potvrdio da se nalazi u restoranu. Ova funkcionalnost povećava sigurnost i sprječava mogućnost lažnih narudžbi.

Prikazan je postupak izrade web aplikacije, od planiranja do implementacije, te su opisane korištene tehnologije. Također, opisane su uloge različitih korisnika aplikacije kao i njihove

funkcionalnosti te je predstavljena i administratorska uloga koja omogućuje izmjenu podataka i statistiku narudžbi.

Ova web aplikacija pruža praktično i korisno rješenje za restorane koji žele olakšati proces narudžbe i posluživanja hrane i pića. Implementirane funkcionalnosti omogućuju učinkovit rad i smanjenje mogućnosti grešaka te time poboljšavaju ukupno iskustvo korisnika.

LITERATURA

[1] Menufy: https://restaurant.menufy.com 20.01.2023.

[2] GrubHub: https://www.grubhub.com/ 20.01.2023.

[3] Waiterio: https://www.waiterio.com/hr/ 13.09.2022.

[4] Progressive Web Apps: https://web.dev/learn/pwa/progressive-web-apps/ 22.01.2023.

[5] Angular minds: https://www.angularminds.com 22.01.2023.

[6] HTML dokumentacija: https://developer.mozilla.org/en-US/docs/Web/HTML 10.09.2022.

[7] CSS: https://developer.mozilla.org/en-US/docs/Web/CSS 23.01.2023.

[8] JavaScript dokumentacija: https://developer.mozilla.org/en-US/docs/Web/JavaScript 10.09.2022.

[9] Bootstrap dokumentacija: https://getbootstrap.com/docs/4.6/getting-started/introduction/ 09.09.2022.

[10] Vue.js dokumentacija: https://v3.vuejs.org/guide/introduction.html 10.09.2022.

[11] Element-Plus dokumentacija: https://element-plus.org/en-US/guide/design.html 05.11.2022.

[12] Pinia dokumentacija: https://pinia.vuejs.org/api/ 04.10.2022.

[13] Axios: https://axios-http.com/docs/intro 22.01.2023.

[14] PHP dokumentacija: https://www.php.net/manual/en/index.php 22.09.2022.

[15] W3Schools, SQL: https://www.w3schools.com/sql/ 10.09.2022.

[16] MySQL dokumentacija: https://dev.mysql.com/doc/ 22.09.2022.

[17] Visual Studio Code dokumentacija: https://code.visualstudio.com/docs 10.09.2022.

[18] QR kodovi: https://qrcode.tec-it.com/ 12.01.2023.

SAŽETAK

Na temu “Web aplikacija za kreiranje narudžbe u restoranu” izrađena je web aplikacija koja omogućuje korisnicima jednostavno naručivanje hrane i pića u restoranu putem mobilnog uređaja. Aplikacija je izrađena korištenjem Vue.js tehnologije za frontend, PHP-a za backend i MySQL baze podataka za pohranu podataka. U aplikaciji se koriste QR kodovi za jednostavnu prijavu putem skeniranja za narudžbu hrane i pića. Postoji mogućnost prijave i različitih razina pristupa za korisnike - goste restorana, konobare, kuhare, administratore i generatore. Aplikacija je testirana na različitim uređajima i preglednicima te je potvrđena jednostavnost naručivanja hrane i pića na željeni način. Implementirane su osnovne funkcionalnosti za prikaz statistike narudžbi, označavanje narudžbi kao poslužene sa strane konobara i kuhara te dodavanje i brisanje konobara i kuhara te isto tako i artikala od strane administratora zajedno sa ažuriranjem. Cilj ovog rada je bio izraditi praktičnu primjenu modernih web tehnologija (poput korištenog Vue.jsa) za poboljšanje korisničkog iskustva u restoranu i olakšavanje procesa narudžbe hrane i pića, a aplikacija je uspješno ostvarila svoj cilj.

Ključne riječi: Pinia, restoran, Vue.js, QR kod, web aplikacija.

ABSTRACT

Title: Web application for creating orders in a restaurant

A web application enables users to easily order food and drinks in a restaurant through a mobile device. The application was developed using Vue.js technology for frontend, PHP for backend, and MySQL database for data storage. QR codes are used in the application for easy login and ordering of food and drinks. There is a login feature and different levels of access for users, including restaurant guests, waiters, chefs, administrators, and generators. The application was tested on different devices and browsers, and the simplicity of ordering food and drinks in a desired manner was confirmed. Basic functionalities for displaying order statistics, marking orders as served by waiters and chefs, adding, deleting, and updating waiters, chefs, and items by administrators were implemented. The aim of this work was to create a practical application of modern web technologies (such as used one Vue.js) to improve the user experience in a restaurant and facilitate the process of ordering food and drinks, and the application successfully achieved its goal.

Keywords: Pinia, QR code, restaurant, Vue.js, web application.

PRILOZI

Aplikacija je dostupna online:

https://toni-web.com/

https://toni14nexe.000webhostapp.com/

Frontend dio projekta s izvornim kodom dostupan je na GitLab-u:

https://github.com/toni14nexe/toni14nexe.github.io

Backend dio za povezivanje na bazu podataka dostupan je na GitLab-u:

https://github.com/toni14nexe/backend-final-paper

Baza podataka i QR kodovi dostupni su na GitLab-u:

https://github.com/toni14nexe/database-final-paper

ŽIVOTOPIS

Toni Kolarić rođen je 20. prosinca 1998. godine u Našicama te je odrastao u Martinu, malom mjestu smještenom uz samo Našice. Pohađao je osnovnu školu Dore Pejačević u Našicama, a nakon čega upisuje srednju školu Izidora Kršnjavoga u istom gradu gdje je stekao zvanje tehničara za elektrotehniku. Uz posao konobara u lokalnom kafiću u Našicama, upisao je stručni studij informatike na Fakultetu Elektrotehnike, Računarstva i Informacijskih tehnologija u Osijeku. Tijekom posljednje godine studija zapošljava se u informatičkoj tvrtki Barrage d.o.o. kao frontend software developer u kojoj i trenutno radi.