

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Luka Horvat

Elektronsko naročanje v restavraciji

DIPLOMSKO DELO

VISOKOŠOLSKI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTORICA: doc. dr. Mira Trebar

SOMENTOR: as. dr. David Jelenc

Ljubljana, 2021

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge: Elektronsko naročanje v restavracij

Besedilo teme diplomskega dela študent prepíše iz študijskega informacijskega sistema, kamor ga je vnesel mentor. V nekaj stavkih bo opisal, kaj pričakuje od kandidatovega diplomskega dela. Kaj so cilji, kakšne metode uporabiti, morda bo zapisal tudi ključno literaturo.

Na tem mestu zapišite, komu se zahvaljujete za izdelavo diplomske naloge. Pazite, da ne boste koga pozabili. Utegnil vam bo zameriti. Temu se da izogniti tako, da celotno zahvalo izpustite.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Uporabljene tehnologije in programske opreme	3
2.1	Vue.js	3
2.1.1	Kaj je reaktivnost?	4
2.1.2	Delovanje reaktivnost v Vue	4
2.1.3	Knjižnice in dodatki	5
2.2	Python/Flask	6
2.3	MySQL/Toad DataModler	7
2.4	Viusal Studio	7
2.5	XAMPP	8
2.6	Git Hub	8
3	Nevem	9
3.1	Problematika	9
3.2	Ideja	10
3.3	Analiza trga	10
4	Struktura in delovanje aplikacije	11
	Literatura	13

Seznam uporabljenih kratic

kratica	angleško	slovensko
CA	classification accuracy	klasifikacijska točnost
DBMS	database management system	sistem za upravljanje podatkovnih baz
SVM	support vector machine	metoda podpornih vektorjev

Povzetek

Naslov: Elektronsko naročanje v restavraciji

Avtor: Luka Horvat

Slovenija velja za državo z veliko število restavracij, vendar le malo iz med njih uporablja napredne sisteme naročanja kot npr. ena izmed večjih verig s hitro prehrano McDonalds. Obstajali so projekti, ko so naredili aplikacije, ki bi/so vpeljale nekaj tehnologije v restavracije vendar v Sloveniji niso uspeli. Eden ključnih razlogov zakaj niso uspeli je bilo sabotiranje s strani natakarjev, saj so misli da bo tehnologija zamenjala njegove službe. V zavedanju njihovih problemov sem se odločil narediti diplomski nalogo na to temo. Rezultat je aplikacija, ki jo je mogoče uporabiti samo kot pregled hrane in pijače, ki jo ponuja restavracija ali pa kot sistem za naročanje. V diplomski nalogi najprej opišem kako sem prišel do ideje, sam razvoj aplikacije in njeno končno delovanje. Predstavim vse težave na katere sem naletel med samim razvojem. Na koncu sem pripravil tudi primerjavo s konkurenco na trenutnem trgu oziroma z aplikacijo, ki vem da ni uspela. Seveda sem pripravil tudi seznam izboljšav, ki bi mojo aplikacijo lahko pripeljalo med eno izmed boljših.

Ključne besede: Slovenija, naročanje, propadel projekt, spletna aplikacija.

Abstract

Title: Diploma thesis sample

Author: Luka Horvat

This sample document presents an approach to typesetting your BSc thesis using L^AT_EX. A proper abstract should contain around 100 words which makes this one way too short.

Keywords: computer, computer, computer.

Poglavje 1

Uvod

Tehnologije v sedanjem času predstavljajo velik napredek na vseh področjih. Na strani gostinstva ni drastičnih napredkov, saj sem to ugotovil kar na osnovi lastnih izkušen. Kot študent fakultete za računalništvo in informatiko sem opravljal delo v eni izmed restavracij, saj biti študent z dodatnim zaslužkom ni dilema. Opravljal sem delo dostavljavca hrane v eni izmed bližnji restavraciji, hkrati pa opazoval potek dela v kuhinji in strežbi. Kasneje sem napredoval v strežbo in kot programer hitro opazil stvari, ki bi se jih dalo izboljšati. Najbolj me je motilo nezadovoljstvo strank ob veliki zasedenosti restavracije. Preprosto en človek ne uspe postreči več kot eno mizo na enkrat. Zato sem si zamislil sistem za naročanje hrane, ki ne bi bil namenjen zamenjavi ljudi v strežbi, vendar samo kot pregledovalnik (angl. Menu) ali pa naročanju jedi in pijače. Stranka bi bila tista, ki bi se odločila ali želi pri naročanju uporabiti stik z osebo v strežbi ali bi preprosto naročila z uporabo aplikacije na tablici, ki bi bila vedno dosegljiva na vsaki mizi.

Poglavje 2

Uporabljene tehnologije in programske opreme

V diplomski nalogi sem se srečeval predvsem z naslednjimi programskimi jeziki: JavaScript, Python, SQL. Vse podrobnosti pa so opisane v spodnjih poglavjih.

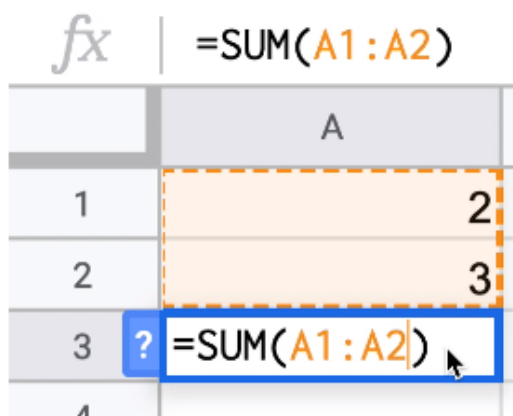
2.1 Vue.js

JavaScript je programski jezik, zaradi katerega so spletne strani postale dinamične in bolj zmogljive. Z njem smo programsko kodo, ki je bila na strežniškem delu, preselili v brskalnik. Tako smo na začetku dobili veliko JavaScript programske kode povezane z različnimi HTML in CSS datotekami brez kakršne koli formalne organizacije (poznano kot «script»). Zaradi tega smo razvijalci začeli uporabljati JavaScript Frameworks, saj poenostavijo izdelavo aplikacij.

Vue je eden izmed mnogih kot npr. Angular, Ember, React, ... poznan pa je predvsem zaradi enostavnosti za upravljanje in izvajanje testov. Vsem pa je skupna točka reaktivnost, vendar v drugačnem pomenu besede. Gre za to, da se aplikacija postopno prilagaja glede na vrednosti podatkov.

2.1.1 Kaj je reaktivnost?

Reaktivnost [4], je programska paradigma, ki nam omogoča, da se na deklarativni način prilagodimo spremembam. Dober primer reaktivnosti je npr. funkcija SUM, ki jo uporabljamo v Excelu. Slika 2.1 prikazuje primer v Excelu.



Slika 2.1: Primer funkcije SUM v programu Excel

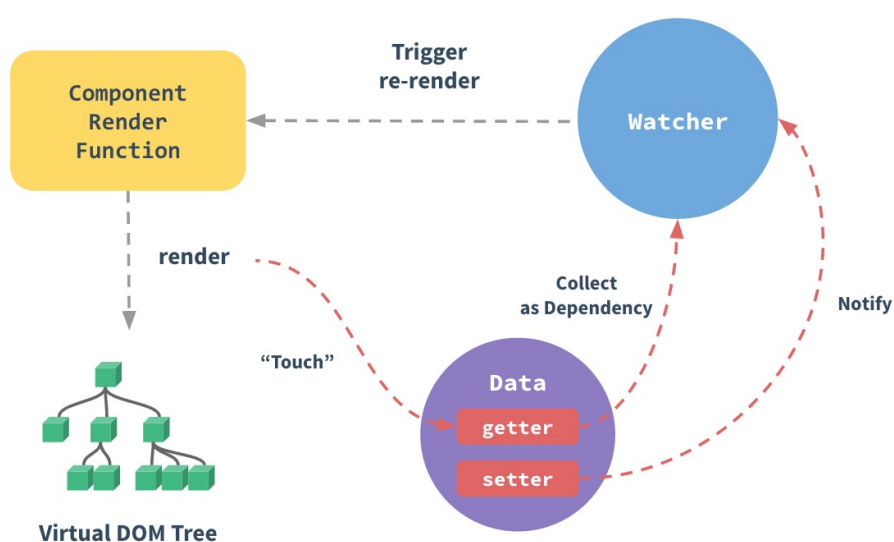
Če vstavimo številko 2 v prvo celico in številko 3 v drugo celico ter izberemo funkcijo SUM teh dveh celic. Kot rezultat dobimo vsoto obeh števil skupaj, kar ni nič posebnega. Vendar če bomo spremenili vrednost prve celice, bo funkcija SUM avtomatsko posodobila skupno vrednost. Tako deluje tudi reaktivnost v aplikacijah za razliko, da je na nek podatek lahko vezanih več funkciji oziroma delov programske kode, ki se ob spremembi njegove vrednosti posodobijo.

2.1.2 Delovanje reaktivnost v Vue

Vue se torej v primerjavi z navadnim JavaScriptom sprehodi skozi podatke in njihove lastnosti (angl. Properties) pretvori v funkciji Getter in Setter, ki sta nevidni uporabniku [3]. Poglejte si sliko 2.2 za lažjo predstavo.

Torej funkcija Getter pokliče instanco Watcher z namenom odvisnosti do

drugih komponent. To pomeni, če je podatek označen kot odvisen (angl. Dependency) bodo nekateri deli programske kode oziroma funkcije poklicane vsakič, ko se spremeni vrednost podatka. Funkcija Setter pa obvesti instanco Watcher, vsakič ko se podatku spremeni lastnost. Ta poskrbi, da se pokliče funkcijo upodabljanja (angl. Render) tiste komponente, ki potem prikaže spremembe v samem pogledu aplikacije.



Slika 2.2: Kateri dialekt uporabljati?

2.1.3 Knjižnice in dodatki

Tako kot vsak programski jezik ima tudi Vue svoje dodatke, ki pomagajo pri razvoju aplikaciji. Spodnje sem uporabljal sam in so vredni omembe.

Vue CLI velja kot standardno orodje za ekosistem Vue [5]. Zagotavlja, da že pri gradnji novega projekta povezuje različne dodatke med seboj. To omogoča razviljacu, da se bolj osredotoči na programiranje in ne na povezovanje njih v celoto. Zadeva izgleda nekako tako, da preko CLI

vmesnika izbereš kakšen projekt želiš. Imaš seveda že privzete nastavitve, vendar omogoča tudi nastavljanje po meri. Sam sem uporabil Vuex, Vue-Router, ESLint in Vuetify.

Vuex je knjižnica za shranjevanje vrednosti v aplikacijah Vue.js [7]. Služi kot centralizirana baza podatkov za vse komponente v aplikaciji.

Vue-Router je uradni usmerjevalnik za Vue.js [6]. Integrira se globoko z jedrom Vue.js, tako da poenostavi izdelavo SPA aplikacij. Usmerjevalki je mišljen v smislu usmerjanja na druge komponente (angl. Component), ki v Vue.js predstavljajo druge poglede, lahko bi rekli podobno kot podstrani.

ESLint je orodje za prepoznavanje in priporočanje o popravkih v programski kodi [9]. Cilje je narediti kodo bolj usklajeno in urejeno, kar pripomore k izogibanju napak.

Vuetify je eden izmed mnogih uporabniških vmesnikov, ki je zgrajen na vrhu Vue.js [8]. Za razliko od drugih vmesnikov je Vuetiy enostaven za učenje z več stotimi komponentami izdelanih po specifikacijah Material Design.

Vue-devtools je zgolj dodatek v brskalniku, ki omogoča lažje sledenje delovanja aplikacije in odpravljanju napak.

2.2 Python/Flask

Flask je eno izmed najbolj popularnih spletno aplikacijskih vmesnikov (angl. Freamwork) [2]. Zasnovan je tako, da omogoča hiter in enostaven začetek z možnostjo razširitve na zapletene aplikacije. V primerjavi z Django spletnim vmesnikom je za enak primer veliko bolj ekspliciten. Flask je prvotno zasnoval in razvil Armin Ronacher kot prvoaprilsko šalo leta 2010. Kljub taki predstavitvi je Flask postal izjemno priljubljen kot alternativa projek-

tom narejenih v Django. Jaz sem ga uporabil za komunikacijo med aplikacijo in podatkovno bazo. Je enostaven za uporabo z veliko podpore na spletu.

2.3 MySQL/Toad DataModler

MySQL je eden od odprtokodnih sistemov za upravljanje s podatkovni bazami, ki za delo s podatki uporablja jezik SQL. Napisan je v programskem jeziku C in C++ in deluje v vseh modernih sistemih npr. Windows, Linux, OS X, ... Prva verzija je bil razvita leta 1995 s strani Michael Widenius in David Axmark. Kratica My izhaja iz imena prve hčerke očeta Michaela.

Toad DataModler je orodje za izdelavo visokokakovostnih podatkovnih modelov. Omogoča izdelavo logičnih in fizičnih podatkovnih modelov, kar pripore k lažjem razumevanju in razvijanju podatkovne baze. Njegova najboljša funkcionalnost je, da lahko generiramo SQL kodo v različne podatkovne sisteme (npr. MySQL, Ingres, Microsoft Azurem, Microsoft Access, Microsoft SQL Server, ...) . Program sem uporabil prav zato, da sem najprej naredil fizični podatkovni model ter nato generiral MySQL kodo, ki sem jo potem samo še uvozil v XAMPP.

2.4 Visual Studio

Visual Studio Code je odprtokodno razvojno orodje, ki ga je razvil Microsoft leta 2015. Namenjeno je razvoju programov v operacijskem sistemu Windows. Omogoča razvoj aplikacij v jeziku C, C++, Python, Visual Basic .NET, JavaScript, ...

Orodje v osnovi omogoča veliko funkcionalnosti, npr. razhroščevanje, popravljanje sintaktičnih napak, avtomatično dopolnjevanje programske kode, izvajanje programske kode v realnem času, dostop do CLI vmesnika, povezavo z GIT, ... Kar ni vključeno v orodje je mogoče dodati z razširitvami, ki se jih brezplačno namesti znotraj orodja. Sam sem si namestil razširitev za Vue.js s katerimi sem pridobil avtomatsko dopolnjevanje kode in ESLint, da

se je nepravilno urejena koda sproti označevala.

2.5 XAMPP

XAMPP je odprtokodni strežnik za razvoj spletnih aplikacij, ki se uporablja lokalno za testne projekte predno postanejo aktivni vsem preko spleta. Ta strežnik za razvoj spletnih aplikacij vsebuje tudi druge pred namestitvene aplikacije, kot so Apache spletni strežnik, MySQL podatkovno bazo, PHP in Perl. Deluje na vseh platformah Linux, Windows in Mac OS. Vse aplikacije se vklaplja preko XAMPP nadzorne plošče, ki jo lahko vidite na sliki ...

PhpMyAdmin je ime za spletni vmesnik, ki omogoča upravljanje z MySQL podatkovnimi bazami. Program je vključen v XAMPP in sicer znotraj aplikacije MySQL. Je zelo močno orodje in predvsem enostavno za uporabo. Sam sem ga uporabil pri testiranju poizvedb in dodajanju podatkov v podatkovno bazo.

2.6 Git Hub

Git Hub je spletna platforma za distribuirano upravljanje s programsko kodo, ki je bila ustanovljena februarja leta 2008 [1]. Njen poudarek je predvsem na hitrosti, integriteti podatkov in vzporednem toku dela. Namenjen je tako za samostojni kot tudi za kolaborantovi razvoj. Začneš tako, da ustvariš repozitoriji, ki predstavlja glavno programsko kodo. Iz tega se lahko naredi več podružnic (angl. Branch), ki predstavljajo kopijo glavne kode na kateri se lahko izvajajo testi oziroma odpravljajo težav. Podružnice potem združuješ in tako glavna programska koda ne vsebuje napak oziroma predstavlja vedno delujoč izdelek. Git Hub Desktop je aplikacija za operacijske sisteme, ki je zelo enostavna. Ponuja vse funkcionalnosti kot na spletu, vendar zaradi uporabe lokalnih orodij veliko bolj priročna. Ko zaključiš delo narediš Commit in Push ter vse spremembe shraniš v repozitoriji na spletu. Tako nikoli ne izgubiš projekta oziroma imaš sinhronizacijo z ostalimi ljudmi na projektu.

Poglavje 3

Nevem

3.1 Problematika

Z izdelavo spletne aplikacije sem želel zagotoviti lažje delovanje restavracij predvsem olajšati delo natakarjem, ki bi se potem lažje osredotočili na kakovost postrežbe pijač in jedi. Izboljšati sem želel čas postrežbe gostov, kar bi prizaneslo tudi k večjemu številu postreženih gostov v nekem časovnem obdobju. Naredil sem analizo, ki si jo lahko pogledate v članku 3.2. Vsekakor ni moja aplikacija namenjena zamenjavi delovnih mest, vendar samo pohitritvi strežbe. To bi vplivalo tudi na zmanjšanje časa, ki ga gost porabi v primeru, da je natakar zelo zaseden. Obstajajo tablice, katere uporabljajo natakarji pri sprejemanju naročil, ki predvsem pomagajo, da si ne rabijo vsega zapomniti in da lahko pobirajo zaporedna naročila. Iz lastnih izkušenj vem, da z uporabo spomina pride hitreje do napak, saj si preprosto ne moreš vsega zapomniti. Težava je tudi v tem, da moraš za vsako pobrano naročilo do blagajne, da naročilo ne čaka in čimprej pride do kuharja oziroma natakarja, ki pripravlja pijače. Po navadi največ pri pobiranju naročil vzamejo gosti, ki se radi premišljujejo zadnje minute, kar ni nič narobe. Ko sem opravljal delo natakarja sem se spraševal zakaj ne bi ta čas raje posvetil strežbi oziroma pripravi naročil, kot pa pobiranju naročil strank, ki ne vedo kaj bi. Ta aplikacija bi še dodatno zmanjšala čas pobiranja naročil, saj natakar včasih

pride prehitro ali pa prepozno, vendar z uporabo aplikacije bi bilo naročanje vedno na voljo.

3.2 Ideja

Aplikacijo sem si zamislil v treh pogledih, in sicer uporabnik oziroma gost, natakar in kuhar. Gost bi imel na voljo pregled vseh pijač in jedi, ki jih restavracija ponuja. S tem bi pripomogel k ekologiji, saj restavracije ne bi več potrebovale papirnatih menijev. Natakar in kuhar bi imela pregled nad naročili. Naročilo bi razdelil v dva dela, da ima tako natakar kot kuhar nadzor samo nad svojimi naročili. S tem bi omogočil, da se lahko kreira statistika za vsakega posebej.

3.3 Analiza trga

ssadasdasd

Poglavje 4

Struktura in delovanje aplikacije

Z izdelavo spletne aplikacije sem želel zagotoviti lažje delovanje v restavracijah predsem na strani natakarjev in kuharjev. Predsem izboljšanju hitrosti postrežbe upam, da s tem ne bi padalo zadovolstvo strank saj interakcija med ljudmi je vsekakor nekaj bolj pristnega kot s tehnologijo. Vendar celoten svet strmi v razvoj

Literatura

- [1] Github. Dosegljivo: <https://en.wikipedia.org/wiki/GitHub>. [Dostopano: 01. 11. 2020].
- [2] Why is flask good? Dosegljivo: <https://www.fullstackpython.com/flask.html>. [Dostopano: 01. 11. 2020].
- [3] Evan You (ustanovitelj). How changes are tracked? Dosegljivo: <https://vuejs.org/v2/guide/reactivity.html#How-Changes-Are-Tracked>. [Dostopano: 31. 10. 2020].
- [4] Evan You (ustanovitelj). Reactivity in depth. Dosegljivo: <https://v3.vuejs.org/guide/reactivity.html#what-is-reactivity>. [Dostopano: 31. 10. 2020].
- [5] Evan You (ustanovitelj). Vue cli overview. Dosegljivo: <https://cli.vuejs.org/guide/>. [Dostopano: 01. 11. 2020].
- [6] Evan You (ustanovitelj). Vue router. Dosegljivo: <https://router.vuejs.org/>. [Dostopano: 01. 11. 2020].
- [7] Evan You (ustanovitelj). What is vuex? Dosegljivo: <https://vuex.vuejs.org/>. [Dostopano: 01. 11. 2020].
- [8] John Leider (ustanovitelj). Why vuetify? Dosegljivo: <https://vuetifyjs.com/en/introduction/why-vuetify/>. [Dostopano: 01. 11. 2020].

- [9] Nicholas C. Zakas (ustanovitelj). About eslint. Dosegljivo: <https://eslint.org/docs/about/>. [Dostopano: 01. 11. 2020].