FRAMEWORK

Uvod

* 1. Problem

Tehnologije v sedanjem času predstavljajo velik napredek na vseh področij. 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, hkrati pa opazoval potek dela, v kuhinji in strežbi, v eni izmed bližnji restavraciji. 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 želijo pri naročanju uporabiti stik z osebo v strežbi ali bi preprosto naročile z uporabo aplikacije na tablici, ki bi bila vedno dosegljiva na vsaki mizi .

1. Uporabljene tehnologije in programske opreme

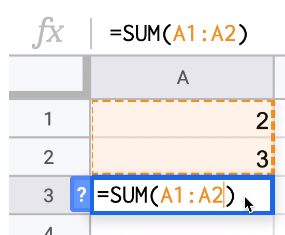
V diplomski nalogi sem se srečeval predvsem z naslednjimi programskimi jeziki: JavaScript, Python, SQL. Vse podrobnosti pa so opisane v spodnjih poglavjih.

* 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 »<sript></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 npr. ko uporabnik klikne na gum s tem sproži dogodek (ang. »event«) in to posledično vodi do posodobitve. reagira pa je po tudi po svoji reaktivnosti, ki omogoča avtomatsko sinhronizacijo ob spremembi vrednosti ali lastnosti JavaScript modela/ov. To se da bolje razložiti s pomočjo spodnje slike.

1. Kaj je reaktivnost? https://v3.vuejs.org/guide/reactivity.html#what-is-reactivity

Reaktivnost 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.

Če vstavitve številko 2 v prvo celico in številko 3 v drugo celico in vnesete funkcijo SUM boste dobili takoj rezultat, kar ni nič posebnega. Vendar če boste spremenili vrednost prve celice, bo funkcija SUM avtomatsko posodobila vrednost.

1. Delovanje reaktivnost v Vue

Vue se torej v primerjavi z navadnim JavaScriptom sprehodi skozi podatke in njihove lastnosti (angl. Properties) pretvori v »getter/setters« funkciji, ki sta nevidni uporabniku. Z uporabo »vue-devtools« vmesnika si zadevo lažje predstavljamo. »Getter« funkcija pokliče »Watcher« z namenom odvisnosti do drugih komponent. To pomeni če je podatek označen kot »dependency« to pomeni, da bodo nekateri deli programske kode/funkcije poklicane vsakič, ko se spremeni vrednost podatka. »Setter« funkcija obvesti »watcher« instanco, 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.

https://deepsource.io/blog/reactivity-in-vue/

https://vuejs.org/v2/guide/reactivity.html#How-Changes-Are-Tracked



* 1. Python/Flask

Flask je eno izmed najbolj popularnih spletno aplikacijski FRAMEWORK. Zasnovan je tako, da omogoča hiter in enostaven začetek z možnostjo razširitve na zapletene aplikacije. Uporabil sem ga predvsem po priporočilo prijatelja. Je enostaven za uporabo z veliko podporo na spletu.

* 1. MySQL/Toad DataModler
  2. Visual Studio
  3. XAMPP
  4. Git Hub

1. Struktura in delovanje aplikacije
   1. Baza
   2. Strežnik – nadzor/komunikacija
   3. Strežnik - aplikacija
2. Pogledi aplikacije
   1. Gost
   2. Natakar
   3. Kuhar
3. Izboljšave in ugotovitve
4. Elektronsko naročanje se dan danes pojavlja na vsakem področju, saj ne glede na to ali ga potrebujemo ali ne nas sama tehnologija vodi v to smer. Sam sem se odločil za elektronsko naročanje v gostinstvu, in sicer zaradi lastnih izkušenj. Kot študent sem moral tudi sam služiti denar, zaradi želje ne obremenjevanja svojih staršev z denarnimi sredstvi. Delal sem kot natakar v eni izmed bližnjih gostiln. Tam sem ugotovil koliko stvari bi se delo bolje optimizirati, da bi stranke bile še bolj zadovoljne in dobile v svoje roke toplejšo hrano oziroma bi bilo celotno doživetje med stranko in natakarjem še boljše. Rešitev sem videl v elektronskih tablicah, saj bi jih lahko uporabili na več načinov. Gostinci bi jih lahko uporabili samo kot pripomoček oziroma meni jedi in pijač ali pa kot sistem za naročanje ob veliki zasedenosti natakarja.

V Sloveniji imamo čez 1000 restavracij in gostilen, vendar le zelo malo iz med njih uporablja napredne sisteme naročanja kot npr. ena izmed večjih verig s hitro prehrano McDonalds. V Sloveniji so obstajal »startup« projekt z ravno takšno idejo kot jo imam jaz. Projekt ni uspel zaradi sabotiranja natakarjev, saj so misli da bodo s tem izgubili svojo službo.

Diplomsko delo zajema Zato v diplomski nalogi najprej predstavim od kje izvira ideja za to tematiko in kakšen je moj prvotni cilj. Potem bom opisal orodja, programske jezike in knjižnice , ki sem jih uporabljal. Predstavil bom podatkovni model in samo strukturo spletne aplikacije. Nadaljeval z podrobnim opisom delovanje aplikacije in naredil primerjavo z že obstoječimi aplikacijami.

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.