# Univerzitet u Beogradu Elektrotehnički faultet

MASTER RAD

Realizacija veb aplikacije za pronalaženje majstora „Majstor“

# Mentor: Kandidat:

dr. Marija Punt, vanr. prof. Nikola Krstić 2022/3014

# Beograd, 2024.

Mojoj majci, ocu i sestri, Koji su mi je podarili ljubav i nežnost u životu.

***Sadržaj***

1. [Uvod 1](#_bookmark0)
2. [Postojeće rešenje i predlog novog 2](#_bookmark1)
   1. [Postojeće rešenje 2](#_bookmark2)
   2. [Predlog novog rešenja 3](#_bookmark5)
3. [Tehnologije korišćene za izradu veb aplikacije 4](#_bookmark7)
   1. [JavaScript 4](#_bookmark9)
   2. [Angular JS 6](#_bookmark10)
   3. [Express.js 10](#_bookmark16)
   4. [PostgreSQL 11](#_bookmark18)
4. [Opis rada aplikacije 12](#_bookmark20)
   1. [Registrovani korisnici 12](#_bookmark21)
   2. [Administrator sistema 12](#_bookmark22)
   3. [Prijavljivanje postojećih korisnika 13](#_bookmark23)
   4. [Registracija korisnika 14](#_bookmark27)
   5. [Promena korisničkih podataka 15](#_bookmark28)
   6. [Predlozi 15](#_bookmark30)
   7. [Jela 16](#_bookmark31)
   8. [Sastavljanje jelovnika 17](#_bookmark35)
   9. [Promena jezika 18](#_bookmark36)
   10. [Porudžbine svih zaposlenih 19](#_bookmark38)
   11. [Narudžbine 20](#_bookmark43)
   12. [Svi korisnici 22](#_bookmark49)
5. [Implementacija sistema 23](#_bookmark52)
   1. [Arhitektura sistema 23](#_bookmark53)
   2. [Početni ekran 24](#_bookmark56)
   3. [PDF preuzimanje 25](#_bookmark59)
   4. [Dvojezičnost 25](#_bookmark61)
   5. [Slanje email-a 26](#_bookmark62)
   6. [Draft 26](#_bookmark64)
   7. [Cron job 27](#_bookmark65)
6. [Zaključak 28](#_bookmark67)

[Spisak 30](#_bookmark68)

# Uvod

Ovaj diplomski se bavi projektovanjem internet aplikacije za naručivanje hrane sa nazivom *FoodApp*. Aplikacija je namenjena firmama koje naručuju hranu za svoje zaposlene i ima za cilj da olakša celokupan postupak naručivanja hrane.

Cilj ovog rada je realizacija internet aplikacije koja će korisnicima skratiti vreme naručivanja hrane i svesta ga na vreme manje od jednog minuta. Aplikacija će smanjiti utrošeno vreme administratora koji je na nivou cele firme zadužen da sastavi jelovnik po želji zaposlenih, odnosno po jelovniku koji je dobio od ketering firme. Administrator mora da sakupi sve narudžbine zaposlenih, da ih objedini i da pošalje ketering firmi broj naručenih jela po danima i vrsti jela.

Ovaj rad će analizirati postojeća rešenja, sagledati njihove nedostatke, sa ciljem da dođe do optimalnog rešenja prateći najnovije svetske tehnologije. Aplikacija će biti realizovana korišćenjem *Angular* radnog okvira na klijentskoj i *Express.JS* na serverskoj strani i kao grafički korisnički interfejs koristiće se *Visual Studio Code*. Za bazu podataka je korišćen *PostgreSQL* u *PgAdmin 4* grafičkom korisničkom interfejsu.

U glavi dva biće predstavljena postojeća rešenja naručivanja hrane sa svojim dobrim i lošim stranama. U nastavku druge glave biće opisano kako će se navedene mane otkloniti.

Kompletna veb aplikacija je razdvojena u dva projekta, serverski deo projekta koji obezbeđuje usluge i klijentski deo projekat koji zahteva usluge i sadrzi korisnički interfejs. U glavi tri će biti navedene i obrazložene tehnologije koje će se koristiti za oba dela ove aplikacije.

U glavi četiri, u prva dva poglavlja će detaljno biti opisane kategorije korisnika dok će se u nastavku detaljno opisati svaka od funkcionalnosti aplikacije.

U glavi pet biće objašnjene neki od najinteresantnijh delova koda klijentskog, odnosno serverskog dela aplikacije.

U glavi šest će biti dat zaključak, biće predstavljeni izazovi koji su se pojavili prilikom implementacije. Biće dati rezultati testiranja kao i moguća dalja unapređenja aplikacije.

## Postojeće rešenje i predlog novog

U ovoj glavi biće predstavljeno postojeća rešenje za pronalaženje majstora, a nakon toga biće dat predlog novog rešenja.

### Postojeća rešenja

#### Postaj

Mobilna aplikacija [*Postaj*](https://postaj.app/), pruža korisnicima mogućnost pretrage majstora po različitim kategorijama i filterima .Na sajtu *Postaj* postoji opcija da klijenti postave oglas u kojem opisuju šta im je potrebno da se uradi, a zatim se majstori javljaju na te oglase. Iako ovo rešenje omogućava korisnicima da dobiju ponude od različitih majstora, smatram da je obrnuti pristup, gde klijenti biraju majstore na osnovu njihovih prethodnih radova, iskustva, i recenzija, logičniji i efikasniji.

Ovaj pristup omogućava klijentima da donesu informisanu odluku na osnovu stvarnog rada i reputacije majstora, umesto da se oslanjaju na ponude koje mogu varirati u kvalitetu i ceni. Na taj način, klijenti imaju veću kontrolu nad izborom, što može povećati njihovo zadovoljstvo uslugom. Takođe, majstori koji su uložili trud u izgradnju portfolija i održavanje visokih ocena mogu biti bolje nagrađeni za svoj rad, jer će privući više klijenata bez potrebe za natjecanjem u formi nadmetanja za posao.

Ovaj model takođe može motivisati majstore da kontinuirano unapređuju svoje veštine i portfolio, jer će im to direktno povećati šanse za dobijanje posla. S obzirom na to, implementacija ovakvog sistema u tvojoj aplikaciji može doprineti stvaranju kvalitetnijeg i transparentnijeg tržišta usluga, što je korisno za sve strane.

Na sajtu *Postaj* postoji izuzetno korisna opcija u vidu bloga gde majstori mogu deliti svoja iskustva, savete, i preporuke. Ova funkcionalnost ne samo da omogućava majstorima da predstave svoje znanje i stručnost, već pruža i veliku vrednost korisnicima koji su u procesu renoviranja ili popravki u svom domu. Ljudi koji traže majstore mogu pronaći mnoštvo korisnih saveta i informacija koje im mogu pomoći u donošenju odluka ili u samostalnom obavljanju određenih poslova. Ovakva opcija čini sajt informativnim i korisnim za sve posetioce, što predstavlja značajnu prednost u odnosu na druga rešenja na tržištu.

#### Profimajstor

Na sajtu *Profimajstor* primećuje se značajan nedostatak u pogledu transparentnosti, jer klijenti nemaju mogućnost da sami odaberu majstora. Umesto toga, operateri sajta dodeljuju majstora po svom nahođenju, što može dovesti do različitih problema.

Ovakav pristup nije optimalan iz nekoliko ključnih razloga. Prvo, iz sigurnosne perspektive, klijenti ne znaju ko dolazi u njihovu kuću, što može izazvati nelagodu i nepoverenje. Drugo, nedostatak informacija o prethodnim radovima tog majstora, uključujući slike i ocene prethodnih klijenata, dodatno otežava klijentima da procene kvalitet usluge koju će dobiti.

Još jedan problem je što, kada se uđe u određenu kategoriju majstora, kao što je npr. moler, umesto konkretnih informacija o majstorima, dostupna su samo opšta mesta i floskule o tome šta taj majstor radi. Ovaj nedostatak specifičnih i relevantnih informacija dodatno smanjuje transparentnost i otežava klijentima donošenje informisane odluke.

Sve ove nedostatke je moguce iskoristiti kao osnovu za unapređenje svoje aplikacije, omogućavajući klijentima da biraju majstore na osnovu detaljnih profila, ocena, i transparentnog prikaza njihovih prethodnih radova, čime bi se povećalo poverenje i zadovoljstvo korisnika.

#### Pozovi majstora

Sajt [*Pozovi majstora*](https://pozovimajstora.com/) pokazuje poboljšanja u odnosu na prethodno analizirane sajtove *Profimajstor* i *Postaj*. Prvo, za razliku od *Profimajstora*, *Pozovi Majstora* omogućava klijentima da biraju majstore na osnovu njihovih prethodnih radova i ocena, čime se povećava transparentnost i sigurnost. Klijenti imaju pristup informacijama o majstorima, uključujući slike njihovih prethodnih radova, ocene od strane prethodnih klijenata, i detalje o njihovom iskustvu, što im omogućava donošenje informisanih odluka.

Takođe, *Pozovi majstora* rešava i problem koji je prisutan na sajtu Postaj gde klijenti postavljaju oglase, a majstori se javljaju na njih. Umesto toga, na sajtu Pozovi Majstora, klijenti imaju slobodu da sami izaberu majstora na osnovu raspoloživih informacija, što im daje veću kontrolu u procesu izbora.

Međutim, u poređenju sa *Postaj* sajtom, Pozovi Majstora ima jedan nedostatak—nedostatak blogova. Blogovi na *Postaj* sajtu omogućavaju majstorima da dele svoja iskustva i savete, što je veoma korisno za ljude koji su u procesu radova na svom domu. Ovaj nedostatak čini *Pozovi Majstora* manje informativnim i korisnim u tom aspektu.

#### Nadji Majstora

Sajt *Nadji Majstora* pruža klijentima mogućnost da biraju majstore na osnovu ocena i dostupnih informacija, što značajno povećava nivo transparentnosti i sigurnosti. Ovaj pristup omogućava korisnicima da imaju veću kontrolu nad procesom izbora majstora, jer mogu odabrati onog koji najbolje odgovara njihovim potrebama i očekivanjima.

Međutim, postoji značajan nedostatak—klijenti ne mogu videti slike prethodnih radova majstora. Ovaj nedostatak otežava procenu kvaliteta rada majstora, što može smanjiti poverenje i sigurnost prilikom donošenja odluka. Slike prethodnih radova su ključne za klijente da bi mogli vizuelno proceniti da li majstor ispunjava njihove standarde.

Sajt *Nadji Majstora* nudi nekoliko pozitivnih aspekata. Veliki broj kategorija omogućava precizno pronalaženje majstora za specifične zadatke, što je korisno za klijente koji tačno znaju šta traže. Detaljna kategorizacija pomaže u lakoj pretrazi i pronalaženju odgovarajućih majstora.

Međutim, ovaj pristup može imati negativne posledice. Ako je zajednica majstora na sajtu mala, neke od podkategorija mogu ostati prazne, što otežava pretragu i navigaciju za klijente. Previše granulirane kategorije mogu učiniti proces pretrage zamornim.

Pored toga, nedostatak blogova na sajtu *Nadji Majstora* znači da klijenti ne mogu dobiti korisne informacije i savete koji bi mogli pomoći u donošenju odluka. Odsustvo slika prethodnih radova dodatno otežava procenu kvaliteta usluga pre nego što se donese odluka o angažovanju majstora.

### Predlog novog rešenja

Kao najveća mana pokazala se loša responzivnost na malim uređajima koji se sve više koriste. S obzirom na to da se količina podataka koju je potrebno prikazati korisniku, u ovom slučaju su to jela razvrstana po danima i vrstama, ne može smanjiti biće nužno uvesti skroler na stranici. Kako bi se poboljšali preglednost ceo jelovnik će biti podeljen na tabele koje predstavljaju dane u toku nedelje. Unutar tabela kolone će predstavljati vrste jela (supa, glavno jelo, prilog, salata, dezert) i u njima će se nalaziti jela koja korisnici mogu da odaberu. Klikom na neko jelo to jelo će biti odabrano. ([slika 2](#_bookmark6))

Tabela koja se nalazila sa desne strane u kojoj je suma svih naručenih jela za registrovanog korisnika je nepotrebna. Ona će biti izmeštena na posebnu stranicu kojoj će ekskluzivno pravo pristupa imati samo administrator. Korisnici će i dalje imati mogućnost uvida u narudžbine svojih kolega ali kako bi aplikacija bila što preglednija tabela sa narudžbinama ostalih korisnika biće na zasebnoj stranici.



*Slika 2 Predlog novog rešenja*

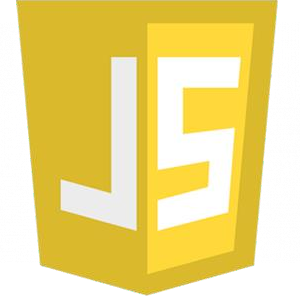
## Tehnologije korišćene za izradu veb aplikacije

U okviru ove glave biće predstavljene tehnologije koje su korišćene za izradu veb aplikacije. Prvo će biti opisan *JavaScript* kao zajednički programski jezik klijentske i serverske strane, nakon toga i njihovi radni okviri *Angular* i *SpringBoot* respektivno, i na kraju baza podataka – *MySQL*.

### JavaScript

Od prvih veb stranica do danas se pristup u njihovom korišćenju uveliko promenio. Korisnici više ne žele samo da čitaju sadržaj kao tekst ispisan u knjigama, nego žele imati uticaja na njega. Zbog toga je kreiran *JavaScript*. Iako mu je prva svrha bila početnicima omogućiti lako dodavanje animacija i interaktivnosti na svoje veb stranice, danas se koristi za izradu vrlo kompleksnih aplikacija. Kako su aplikacije pisane u *JavaScript* programskom jeziku postajale sve kompleksnije, javila se potreba za dobro struktuiranim kodom koje je moguće lako održavati.

Brojne primene dovode do toga da je *JavaScript* trenutno najpopularniji programski jezik na svetu, a to dalje znači da se sa poznavanjem tog jezika otvaraju izuzetne poslovne mogućnosti. Kako se sve veći broj biznisa odlučuje za online poslovanje, mogućnosti dalje rastu, pa se predviđa da će interesovanje za JavaScript, njegova upotreba i relevantnost samo biti u porastu narednih godina ([slika](#_bookmark8) 3).



*Slika 3 JavaScript*

U pitanju je takozvani objektni skriptni jezik, čija je osnovna svrha korišćenja vezana za poboljšanje dinamike određene stranice. Zapravo je stvar u tome da kada se koristi *HTML* kod, njegova uloga je samo da oblikuje određene elemente stranice, kao što su recimo tekst, tabele, linkovi ili forma, te da ih nakon toga uredi, ali ovaj kod nema ulogu da odredi na koji način će se konkretni elementi ponašati. Upravo tada se u čitav procesu uključuje JavaScript, koji ima značajnu ulogu u kreiranju ponašanja ovih elemenata. A kada se on kombinuje sa *HTML* i *CSS* programskim jezicima, tada nastaje dinamički *HTML* (*DHTML*).

Kada se kaže da je neki skriptni jezik objektni, to zapravo znači da osoba koja radi sa njim treba da odredi i tip podataka, ali i da definiše koja će tačno funkcija biti primenjena, a što se mora precizirati na osnovu strukture konkretnih podataka. Takav princip dovodi do toga da struktura određenih podataka ustvari počinje da bude tretirana kao objekat u okviru koga se nalaze i ti podaci i određene funkcije. Korišćenjem ovog programskog jezika se takođe određuju i odnosi između dva objekta.

Ovaj skriptni jezik je sačinjen od, naravno brojnih komandi, koje se moraju očitati u okviru specijalnog programa koji je nazvan Interpreter, takozvani prevodilac. On služi da se te komande i očitaju, a da istovremeno ne budu prevedene u takozvanim mašinski jezik. Tačnije, tada komande bivaju iščitane direktno iz koda, pa zato i kažemo da se programski jezik *JavaScript* izvršava na klijentskoj strani.

Iako je do pre par godina *JavaScript* bio prisutan samo na vebu, danas je situacija značajno drugačija. Danas je osim na vebu prisutan i u razvoju mobilnih aplikacija i kao i u razvoju video igara. On je osmišljen da se koristi kao pomoć ostalim “velikim” programskim jezicima za kreiranje veb aplikacija, međutim sa pojavom *Node.js* moguće je kreirati potpuno funkcionalne veb aplikacije pomoću *JavaScript* programskog jezika koje se brinu i za klijentsku i za serversku stranu.

Smatra se upravo da je jedan od najpopularnijih skriptnih jezika koji se koristi u današnje vreme baš JavaScript, a posebno je dobro to što je on podržan od strane svih onih pregledača koji imaju najčešću primenu (*Mozilla Firefox, Opera, Internet Explorer*), ali takođe i na različitim platformama (*Windows, Linix, Mac OS, Android, iOS*). Osim navedenih veb pretraživača, računskih i mobilnih operativnih sistema, *JavaScript* postaje sve uspešniji i u području operativnih sistema za pametnih satova i narukvica, operativnih sistema za televizije, robotiku. (JavaScript 2022) (What is JavaScript? n.d.)

### Angular JS

*Angular* je *JavaScript* radni okvir za razvoj prednjeg dela (klijentskog dela) koji se koristi za razvoj dinamičkih veb aplikacija za desktop i mobilne uređaje. Prva verzija ovog radnog okvira objavljena je 2012. godine od strane kompanije *Google*.

2016. godine pojavljuje se druga verzija *Angular* okvira. *Angular* je iz temelja redizajniran na osnovu saveta iz prakse, ali i tako da bude u koraku s modernim tehnologijama koje su se pojavile nakon prve verzije.

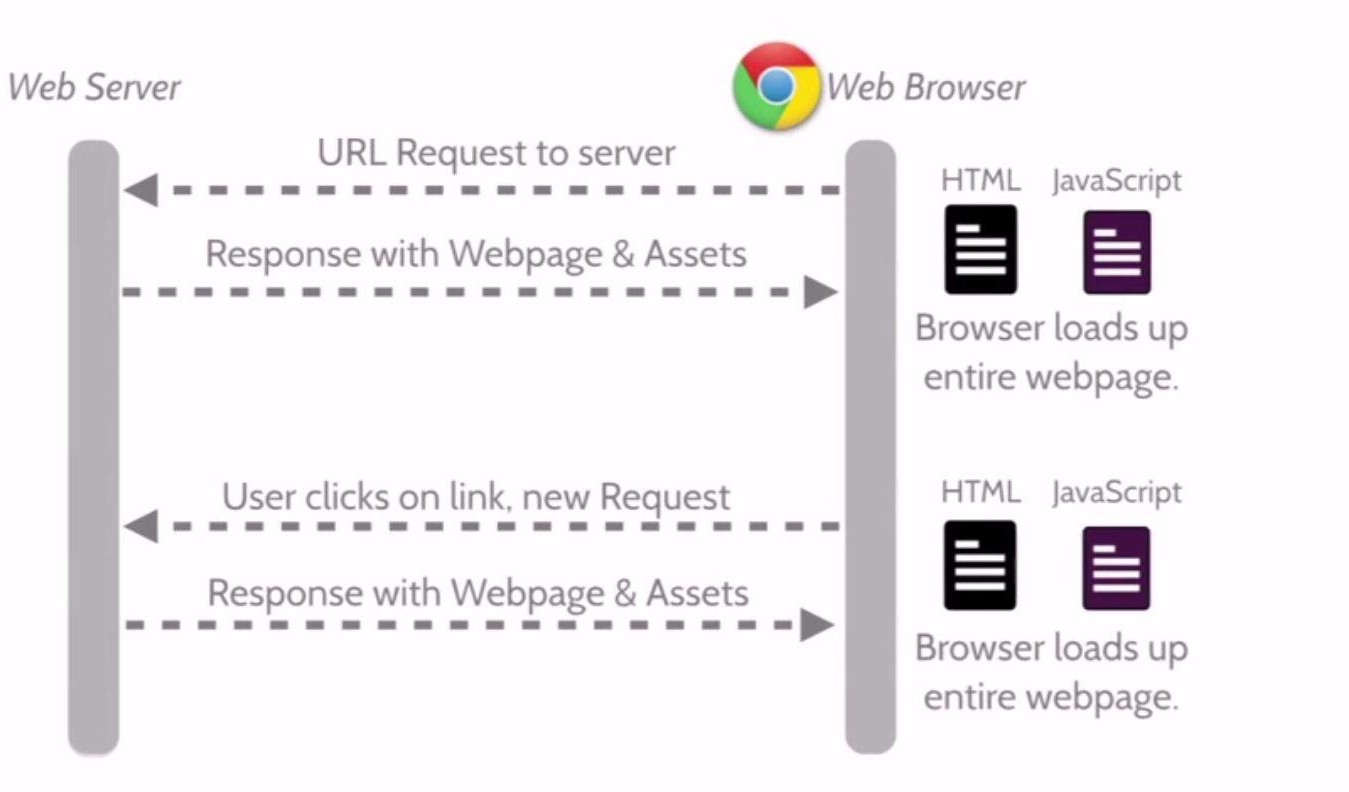
Angular okvir napisan je programskom jeziku *TypeScript*, razvijenom od strane kompanije *Microsoft*. Njegova svrha je nadomestiti nedostatke *JavaScript* jezika, zato se često u literaturi *TypeScript* naziva nadskup *JavaScript* jezika. *TypeScript* nije jezik kojeg pretraživači mogu interpretirati, stoga se *TypeScript* kod prevodi u *JavaScript* kod. *TypeScript* omogućava strogo tipiziranje nad *JavaScript* kodom, to jest možemo deklarisati tipove podataka. Time je omogućena provera nad tipovima podataka prilikom prevođenja koda kako ne bi došlo do greške prilikom izvršavanja koda. Uz strogo tipiziranje i otkrivanje grešaka prilikom prevođenja, najvrednija dopuna *JavaScript* kodu jeste uvođenje mehanizama objektno-orijentiranog programiranja. (AngularJS 2022)

Dobra svojstva prve verzije ovog okvira, poput direktiva i zavisnosti, nasleđena su i poboljšana unutar druge verzije. Mnogi mehanizmi prve verzije su odbačeni radi lakšeg snalaženja prilikom rada s *Angular* okvirom, ali i radi poboljšanja performansi same aplikacije. Oslanjajući se na *TypeScript* sintaksu, svaki se entitet unutar *Angular* okvira definiše pomoću *TypeScript* klase čime je omogućeno jednostavnije pisanje i lakše snalaženje unutar koda. Uz to, *Angular* okvir dolazi s komandnim alatom *Angular-CLI* koji vodi računa o stvaranju kostura same aplikacije. ([slika](#_bookmark11) 4)



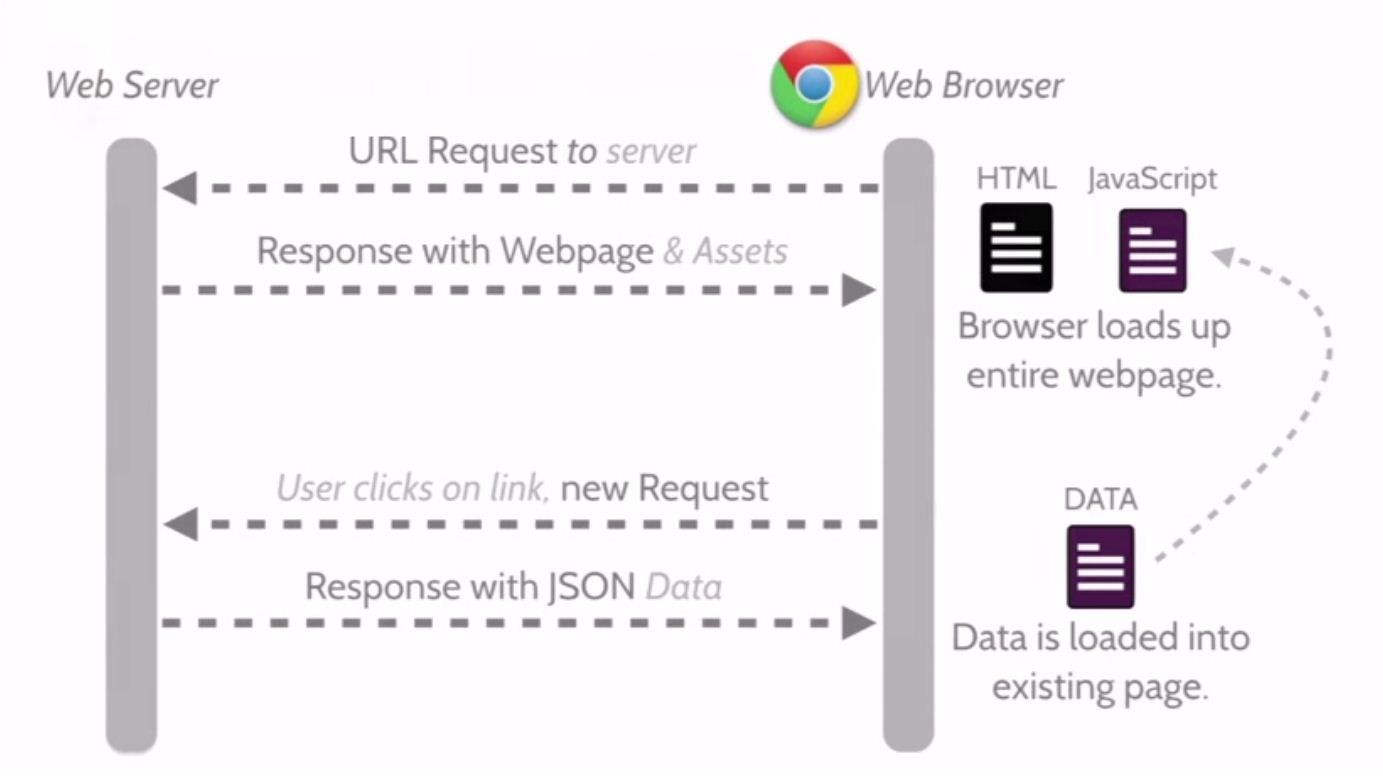
*Slika 4 Angular*

Osnova namena Angular okvira je razvoj jednostraničnih veb aplikacija. Njihova glavna karakteristika jeste prikazivanje sadržaja na jednoj glavnoj stranici što znači da prilikom navigacije korisnika kroz aplikaciju nema potrebe za slanje zahteva serveru za dohvatanje nove *HTML* stranice. *SPA* (*Single Page Applications*) aplikacije se oslanjaju na router. Ruter se sastoje od ruta koje opisuju mesto na kojem bi se trebalo preusmeriti. To mogu biti statične ili dinamičke rute.



*Slika 5 Klijent server komunikacija*

Kod klasične klijent server komunikacije sa jedne strane imamo server dok je sa druge klijent. Klijent predstavlja veb pretraživač u našem slučaju ([slika](#_bookmark12) 5) *Google Chrome*. Kada korisnik u veb pretraživač unese URL, na server se šalje zahtev. Server tada odgovara tako što pošalje veb stranicu, nakon toga korisnik dobije učitanu stranicu. Zatim, kada korisnik klikne na neki link, veb pretraživač na server šalje novi zahtev. Server tada ponovo šalje celu veb stranicu sa svim dodacima, te se sve ponovo učita u veb pretraživač.



*Slika 6 Angular komunikacija sa serverom*

Kada koristimo *Angular* proces je malo drugačiji. Ipak, početak je isti, korisnik unese URL u veb pretraživač pa se zahtev šalje na server. Server tada šalje celu veb stranicu i korisnik dobije učitanu veb stranicu. Kada korisnik klikne na link i želi da pređe na neku drugu stanicu, veb pretraživač šalje novi zahtev na server. Medutim, server sada ne šalje celu stranicu kao pre, nego samo podatke u JSON formatu. *JavaScript* kod stranice tada obraduje dobijene podatke, te na taj način odlučuje šta će korisnik videti na ekranu ([slika](#_bookmark14) 6).

*Angular* je dizajniran sa ciljem da razdvoji *DOM* manipulacije od aplikacione logike. Angular takođe razdvaja klijentsku stranu aplikacije od serverske strane aplikacije. Ovo omogućava rad na razvoju obe strane paralelno, i omogućava ponovnu upotrebu obe strane. On takođe obezbeđuje i strukturu za redosled razvoja aplikacije – od projektovanja korisničkog interfejsa, preko pisanja poslovne logike do testiranja.

Angular implementira MVC arhitekturu radi razdvajanja komponenti za prezentaciju, podatke i logičkih komponenti. Koristeći zavisnost paketa, Angular donosi tradicionalne serverske servise, kao što su kontroleri koji zavise od prezentacionog sloja, na veb aplikacije na klijentskoj strani. Shodno tome, veliki deo tereta na serveru se može smanjiti.

*Angular* biblioteka radi tako što prvo pročita *HTML* strane, koja ima ugrađene dodatne nestandardne tag atribute. *Angular* te atribute interpretira kao direktive da bi vezao ulazne ili izlazne delove stranice za model koji je predstavljen standardnim *JavaScript* promenljivama. Vrednosti tih *JavaScript* promenjivih se mogu ručno podesiti u kodu, ili mogu biti preuzete od statičnih i dinamičnih *JSON* izvora.

U izradi projekta je korišćen i *Angular* materijal koji predstavlja materijalni dizajn za *Angular* komponente. Omogućuje brz razvoj i dizajn korisničkog interfejsa uz moderne komponente. (Angular material n.d.)

Mnogi frejmvorci su napravljeni od strane ljudi iz *Open source* zajednice, Angular je napravljen i održavan od strane inženjera kompanije *Google*. Samim tim možete biti sigurni da radite sa pouzdanim i testiranim kodom na vašem projektu. *McDonald's, AT&T, HBO, Apple, Forbes, Adobe, Nike, a takođe i Microsoft, PayPal* samo su neke od kompanija koje koriste *Angular* i koje mu daju na kredibilitetu.

Izbor grafičkog korisničkog interfejsa za *Angular* je veliki. Razvoj aplikacije u nekoj od popularnih razvojnih okolina (*WebStorm, Visual Studio* ili *Angular IDE*) donosi prednosti kao sto su predlozi programskog koda, trenutni prikaz greški i ostale povratne informacije.

Za *Angular* programere postoji ogroman broj paketa otvorenog koda. Glavni paketi uključuju *NgBootstrap, Angular Google Maps, NgRx, NgTranslate, AngularFire, NgxTextEditor, Angular Material, Ng2 Pdf Viewer* i grafikone. (Npm packages n.d.)Nekim od ovih paketa upravlja službeni *Angular* tim, dok zajednica programera upravlja drugim. Ako želite stvoriti složenu funkcionalnost ili komponentu za svoju aplikaciju, postoji velika verovatnoća da paket već postoji, pa možete iskoristiti postojeće rešenje. (Ayaz 2021)

### SpringBoot

*Node.js* ([slika](#_bookmark17) 8) je višeplatformsko *JavaScript* radno okruženje otvorenog koda za izvršavanje *JavaScript* programskog jezika na serverskoj strani. Istorijski gledano *JavaScript* je primarno korišćen na klijentskoj strani, gde su skripte napisane u *JavaScript* programskom jeziku bile ugrađene u *HTML* stranice, kako bi se izvršile na klijentskoj strani u veb pretraživaču. *Node.js* omogućava da se *JavaScript* koristi za skripte na serverskoj strani koje omogućavaju da se sadržaj dinamičnih veb stranica generiše na serveru pre nego što se pošalje do veb pretraživača korisnika. Zbog toga je *Node.js* postao jedan od osnovnih elemenata paradigme "*JavaScript* svuda" jer omogućava uniformisanje razvoj veb aplikacija u jednom programskom jeziku, bez potrebe da se za skripte na serverskoj strani koristi različit programski jezik. (Node.js 2022)

Korišćenjem *Node.js* postižemo korišćenje jedinstvenog jezika između klijentskog i serverskog dela. To znači da kompletnu aplikaciju možete realizovati pomoću jednog programskog jezika.

*Node.js* je veoma brz. *Google V8 JavaScript engine* i *Node.js* u saradnji, imaju neverovatnu brzinu. Zapravo, nekoliko puta veću od *Ruby* i *Python* skriptnih jezika.



*Slika 8 Node.JS*

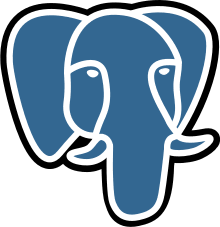
### PostgreSQL

*PostgreSQL* je vrsta objektno-orijentisanih relacionih sistema za upravljanje bazama podataka, pod *open source* licencom. Smatra se jednom od najpouzdanijih baza podataka. Najčešće koristi za veb aplikacije i veb baze podataka. Reklo bi se da je najveći konkurent *MySQL* bazi podataka.

Prva glavna prednost korišćenja *Postgres* baze podataka je to što je *open source* i može se prilagoditi prema zahtevima developera. Ova mogućnost prilagođavanja je izuzetno korisna u razvoju velikih aplikacija. *Postgres* je na tržištu već više od 15 godina i njegova zajednica je u ovom trenutku izuzetno velika, što samim tim znači dobru podršku i pomoć pri rešavanju problema vezanih za rad i uporebu. *Postgres* je izuzetno ekonomičan i ne zahteva mnogo obučavanja

korisnika kako bi se naučilo kako koristiti i programirati za ovu bazu podataka. Takođe, zahtevi za održavanje i podešavanje *Postgres* baze podataka su relativno mali u odnosu na druge sisteme za upravljanje bazama podataka. Dobra stvar u *Postgres* bazi je to što je portabilan i prenosiv sa gotovo svim glavnim platformama i programskim jezicima. Ova baza podataka je idealna za aplikacije namenjene višestrukim platformama. *Postgres* je svetski priznat kao najsigurnija i stabilnija baza podataka. Šanse da se uništi baza podataka su minimalne i čak i ako se baza podataka sruši, postoje načini i funkcije koje vam omogućavaju da obnovite i vratite podatke ([slika](#_bookmark19)

9). Za korisnički grafički interfejs biće korišćen *PgAdmin 4*. (Postgresql 2022, Angular.io n.d.)



*Slika 9 PostgreSQL*

# Opis rada aplikacije

U ovom delu će najpre biti opisane kategorije korisnika a zatim i funkcije koje sistem pruža različitim kategorijama korisnika. Funkcije će biti detaljno opisane uz priložene slike.

### Registrovani korisnici

Registrovani korisnici u projektu predstavljaju zaposlene firme koji naručuju hranu. Registrovani korisnik unošenjem svog korisničkog imena i šifre se prijavljuje na sajt. Takođe postoji i opcija prijavljivanja preko *Google* naloga sa određenim domenom.

Prijavljeni korisnik ima opciju odabira hrane za sledeću nedelju. Za svaki dan ukoliko je on radni dan postoji izlistana jela koja su u ponudi za taj dan (supa, glavno jelo, prilog, salata, dezert) i klikom na određeno jelo ono se odabira, odnosno klikom na polje za potvrdu Neradni dan korisnik ne naručuje hranu za taj dan.

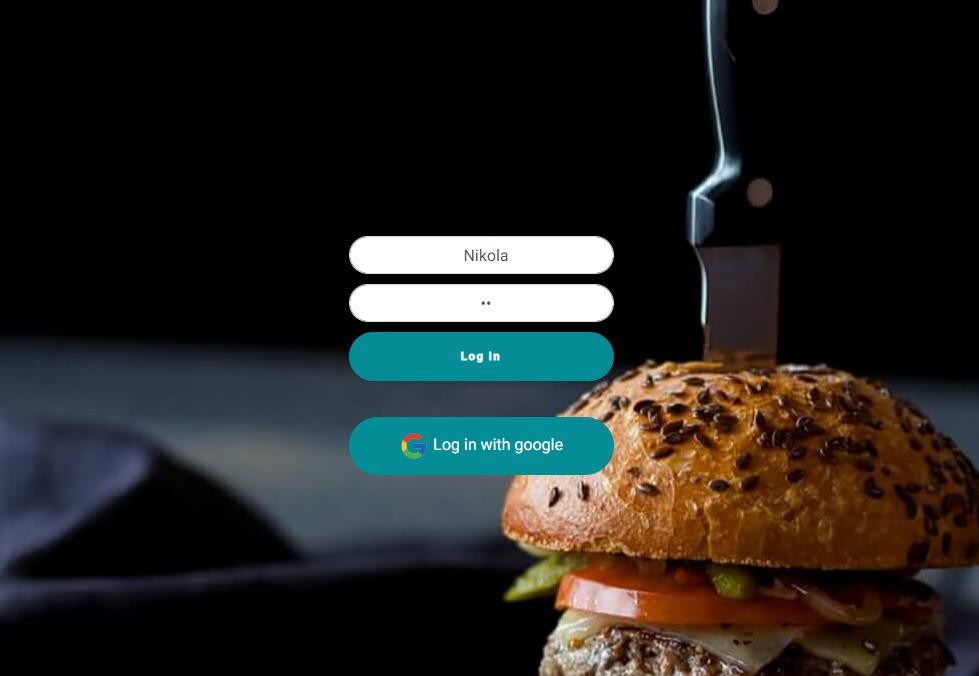
Takođe korisnik ima uvid u tabelarni prikaz izabranih jela ostalih korisnika aplikacije i opciju preuzimanja u PDF verziji date tabele. Korisnik može promeniti svoje lične podatake (korisničko ime, ime, email, slika, lozinka) i dodati odnosno podržati predloge koji se odnose na unapređenje celokunog jelovnika.

### Administrator sistema

Ulogu administratora u projektu ima osoba zaposlena u firmi koja je “posrednik” između zaposlenih i ketering firme. Administrator pored svih funckionalnosti registrovanog korisnika ima ulogu u sastavljanju jelovnika za sledeću nedelju odnosno ima pristup stranici na kojoj su izlistana sva jela gde može da ažurira postojeća ili dodaje nova jela. Administrator ima pristup stranici na kojoj su izlistani svi korisnici i gde može da kreira nove korisnike ili briše postojeće. U odnosu na registrovanog korisnika koji može samo da podržava predloge administrator može i da ih obriše. Pored prikaza narudžbina svakog od zaposlenih, administrator ima mogućnost slanja mejla podsetnika onim korisnicima koji još nisu popunili jelovnik za sledeću nedelju. Takođe ima i pristup stranici u okviru koje se nalazi tabelarni prikaz ukupnog broj jela po danima i opciju preuzimanja u PDF kako bi taj izveštaj mogao da pošalje ketering firmi.

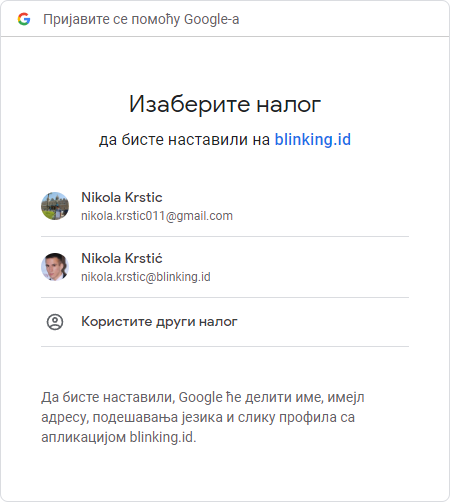
### Prijavljivanje postojećih korisnika

Ova opcija se odnosi na korisnike koji su već registrovani imali oni običnu ili admininstratorsku ulogu u sistemu. Unošenjem korisničkog imena i lozinke, koji se moraju poklapati sa postojećim podacima o korisnicima u bazi podataka, pristupa se nalogu ([slika](#_bookmark24) 10).



*Slika 10 Prijavljivanje*

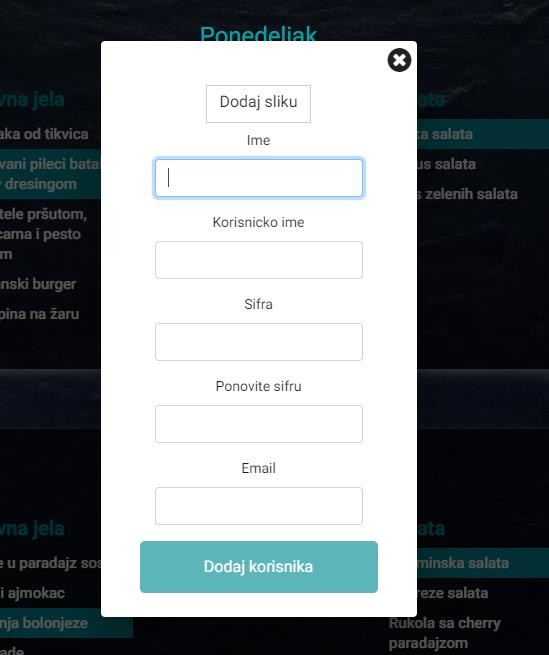
Pored običnog prijavljivanja postoji i opcija prijavljivanja preko *Google* naloga koji se može proveravati da li pripada tačno određenom domenu, u našem slučaju je to domen firme. Klikom na dugme “*Log in with Google*” otvara se modal na kojem je potrebno odabrati i prijaviti se na *Google* nalog ([slika](#_bookmark26) 11).



*Slika 11 Prijavljivanje preko Google-a*

### Registracija korisnika

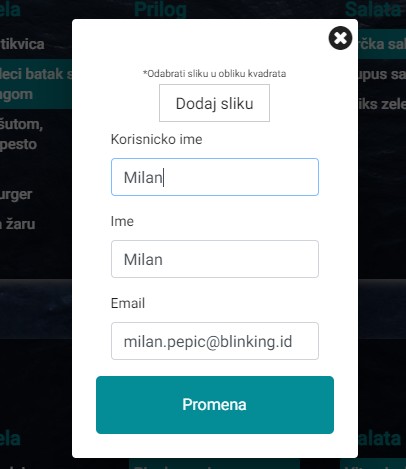
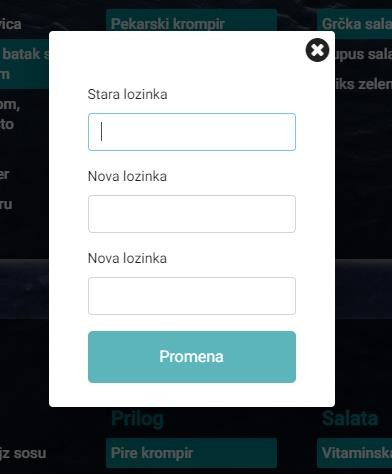
Administrator ima mogućnost dodavanja novog korisnika ([slika 1](#_bookmark26)2) unošenjem njegovih podataka (ime, korisničko ime, lozinka, email, slika). Potreba za registrovanjem novog korisnika u firmi se javlja prilikom puštanja u rad aplikacije ili novo zaposlenog.



*Slika 12 Registraciija korisnika*

### Promena korisničkih podataka

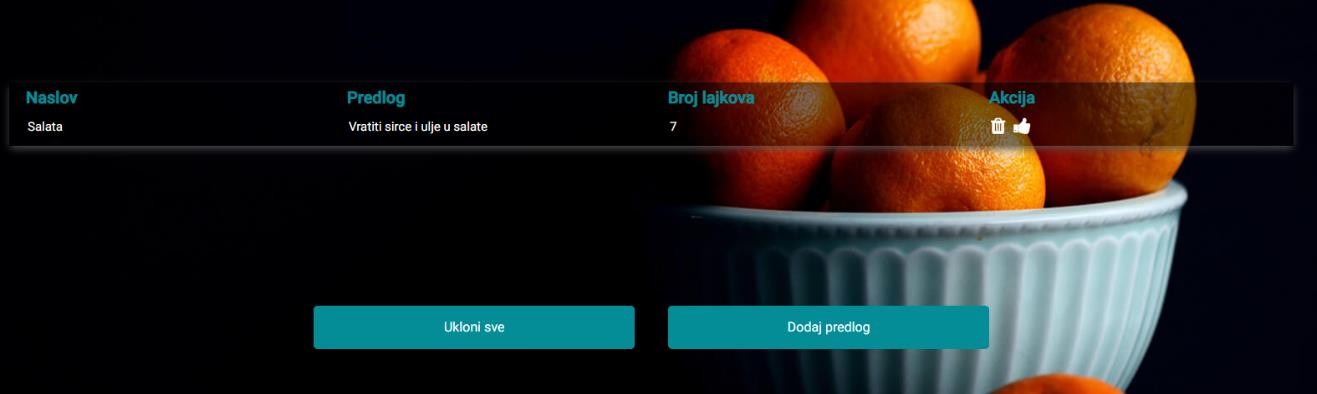
Klikom na profilnu sliku u gornjem desnom uglu aplikacije otvara se padajuća lista. U padajućoj listi zaglavlja postoji mogućnost promene lozinke i korisničkih podataka ([slika 1](#_bookmark29)3).



### Predlozi

*Slika 13 Promena lozinke I korisničkih podataka*

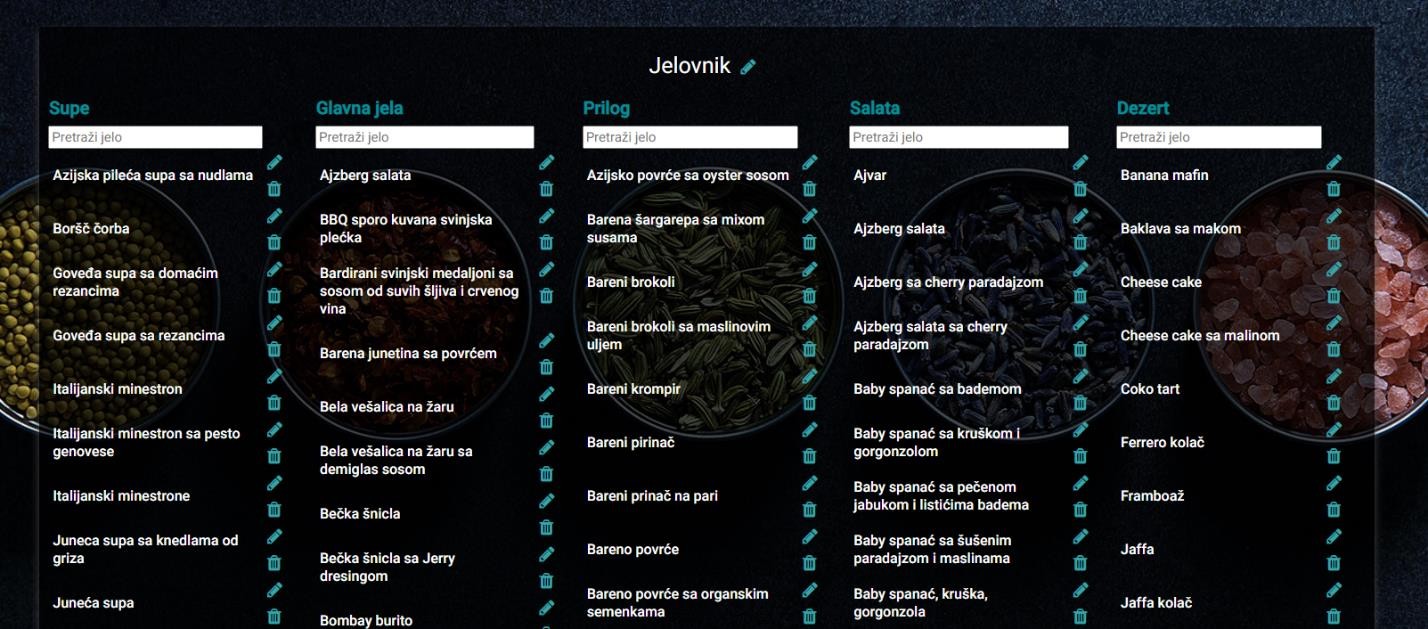
Kako bi zaposleni sve svoje primedbe i pohvale objedinili na jedom mestu postoji stranica predlozi. Registrovani korisnik može da dodaje i podržava predloge, dok administrator ima i mogućnost njihvog brisanja. ([slika 1](#_bookmark29)4).



*Slika 14 Predlozi*

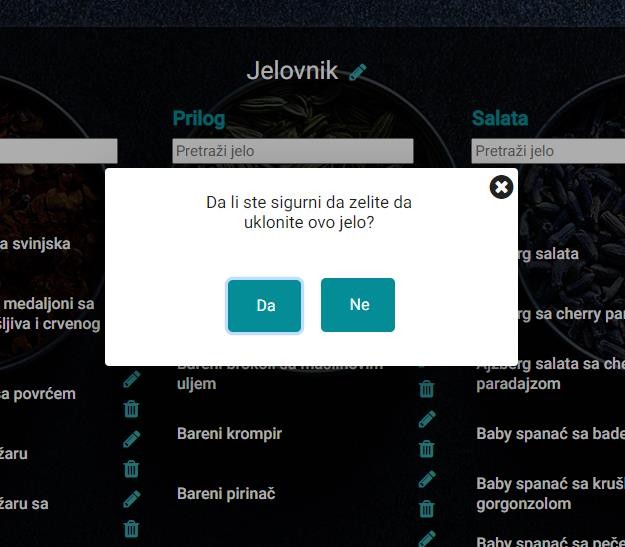
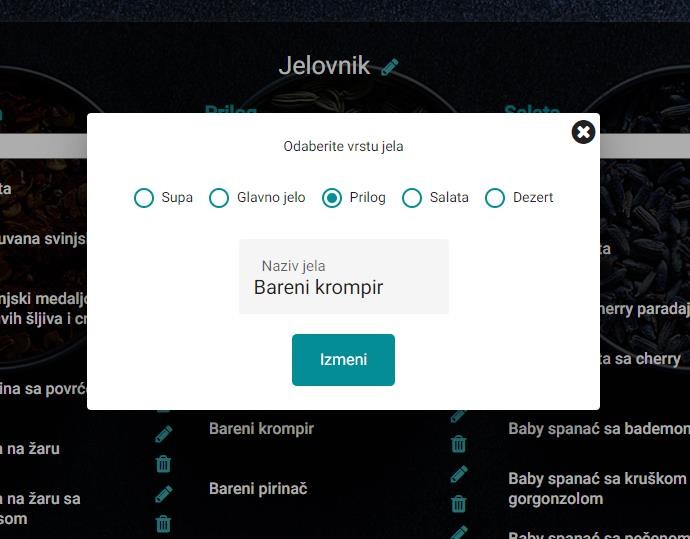
### Jela

Administrator ima pristup stranici na kojoj uređuje jelovnik ([slika 1](#_bookmark33)5). Na toj stranici klikom na olovku pored “Jelovnik” ima mogučnost da obriše odnosno izmeni ime, opis, vrstu datog jela.



*Slika 15 Jelovnik*

Pored svakog od jela administrator klikom na kantu ima opciju brisanja jela([slika 1](#_bookmark32)6). Klikom na olovku dato jelo može izmeniti ([slika 1](#_bookmark32)7).

*Slika 16 Brisanje jela*

*Slika 17 Ažuriranje jela*

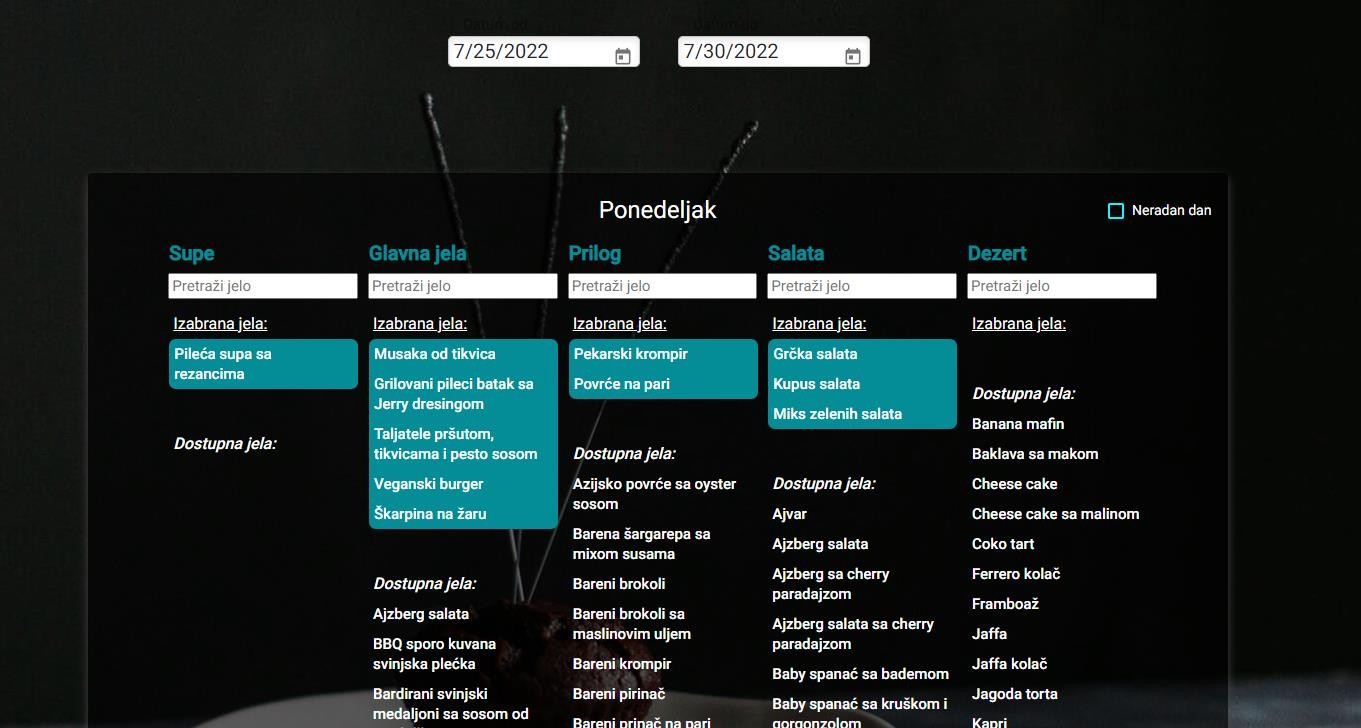
Pored izmene i brisanja jela administrator može i dodavati nova jela. Na vrhu stranice odabirom vrste jela, unošenjem naziva jela i klikom na dugme “Dodaj“ u bazu podataka će se uneti novo jelo ([slika 1](#_bookmark34)8).



*Slika 18 Dodavanje jela*

### Sastavljanje jelovnika

Administrator je zadužen za pravljenje jelovnika za narednu nedelju odnosno za vremenski period koji sam bira. Odabirom prvog datuma drugi datum će se automatski ažurirati na kraj nedelje to jest postaviti na petak date nedelje. Data dva datuma označavaju početak i kraj date radne nedelje ([slika 1](#_bookmark34)9).



*Slika 19 Sastavljanje jelovnika*

U slučaju da administrator smanji opseg nedelje (na primer ponedeljak - četvrtak) petak će promeniti boju u sivu, a korisnicima koju budu popunjavali jelovnik za datu nedelju biće naznačeno da je petak neradan. Neradni dani se takođe mogu i ručno oznaciti klikom na polje za potvrdu u gornjem desnom uglu određenog dana.

Svaki od dana poseduje zasebnu tabelu koja se sastoji od pet kolona koje predstavljaju vrste jela. Na vrhu svake od kolona moguće je pretraživati datu vrstu jela. Klikom na jelo ono se odabira za jelovnik tog dana. Klikom na dugme “Sačuvaj“ na dnu ove stranice jelovnik za datu nedelju će se sačuvati nakon čega će se svim korisnicima poslati mejl obaveštenja da mogu da popune jelovnik za sledeću nedelju.

### Promena jezika

Na dnu padajuće liste, koja se otvara klikom na profilnu sliku u zaglavlju stranice, nalaze se dve zastave, klikom na jednu od njih, britansku odnosno srpsku možete dinamički promeniti jezik veb aplikacije na engleski odnosno srpski respektivno. ([slika](#_bookmark37) 20)

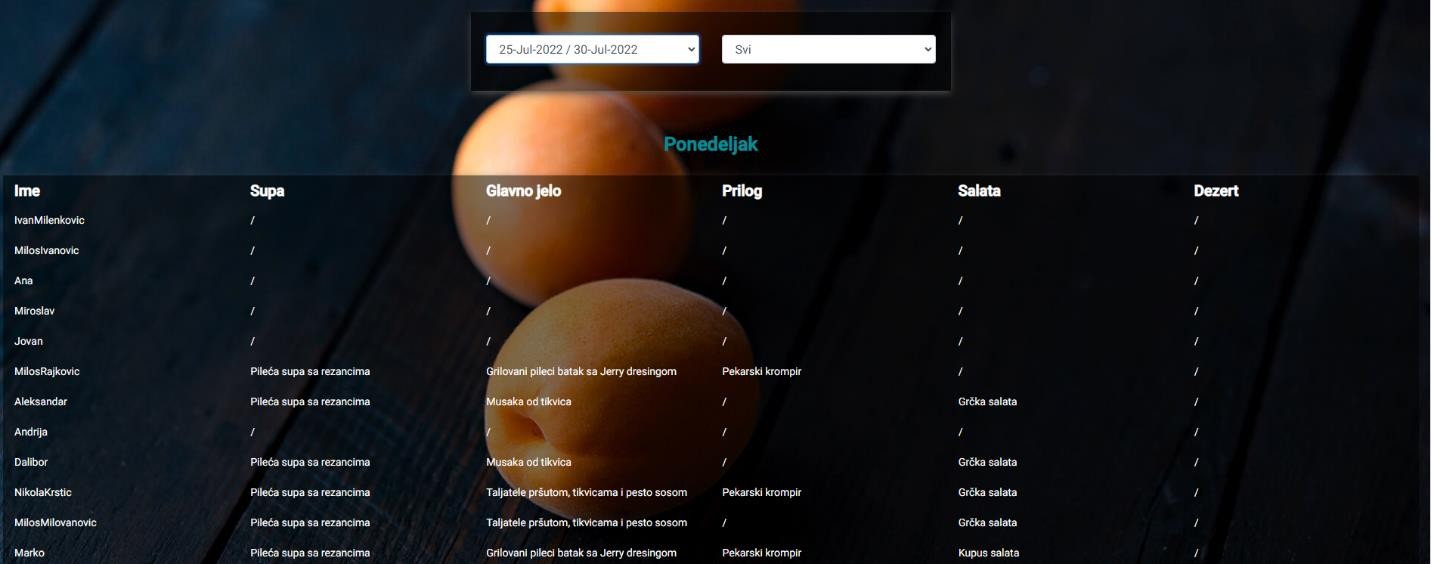


*Slika 20 Promena jezika*

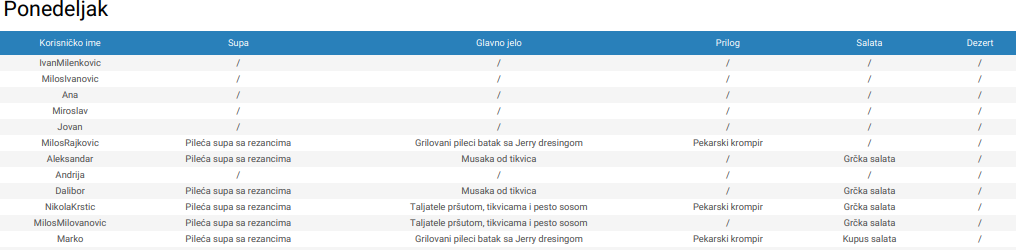
### Porudžbine svih zaposlenih

Stranici „Porudžbine svih zaposlenih“ imaju svi korisnici pristup. Iz prve padajuće liste mogu da odaberu neku od prethodnih nedelja, dok iz druge padajuće liste mogu odabrati tačno određeni dan ([slika](#_bookmark39) 21).

Na dnu stranice nalazi se dugme “Sačuvaj PDF”, klikom na to dugme PDF prikaz narudžbina biće sačuvan([slika 2](#_bookmark41)2).



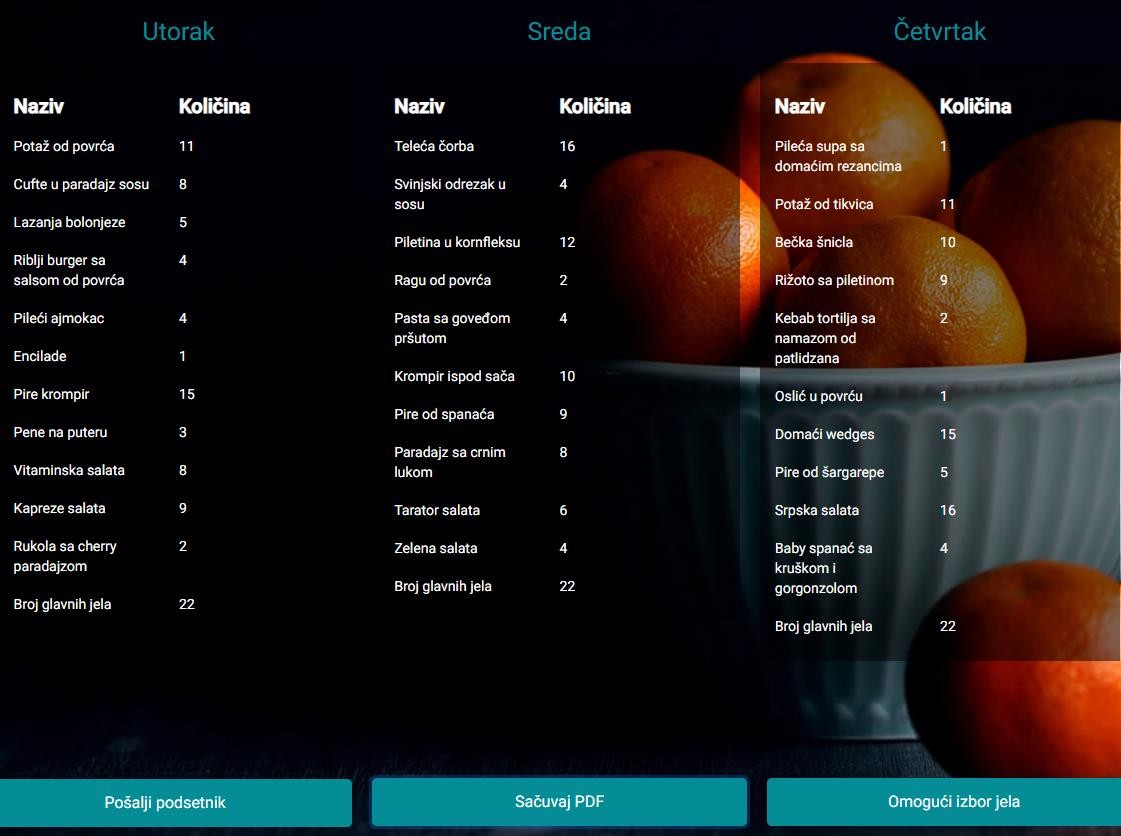
*Slika 21 Porudzbine svih zaposlenih*



*Slika 22 Pdf prikaz porudzbina svih zaposlenih*

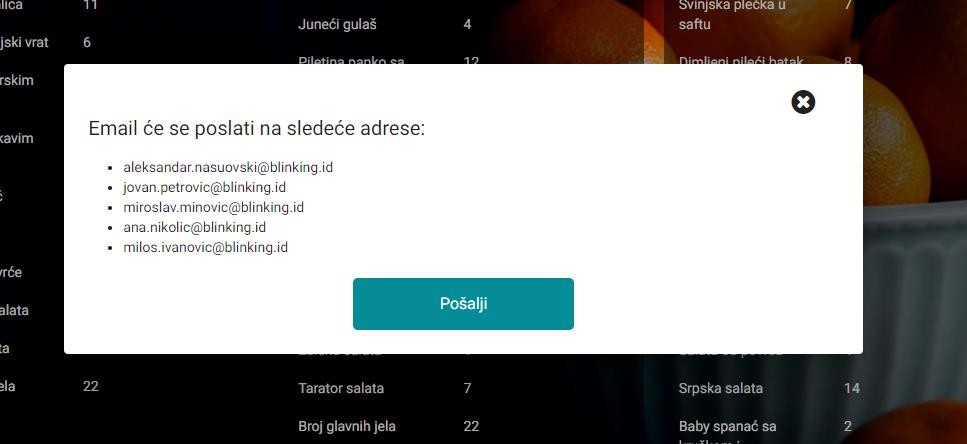
### Narudžbine

Stranici Narudzbine pravo pristupa ima samo administrator i na njoj može videti koliko je kojih jela naručeno po tabelama odnosno danima tekuće nedelje. U padajućoj listi na vrhu strnici može odabrati određeni dan čime će se prikazati samo tabela za taj dan ([slika 2](#_bookmark44)3).



*Slika 23 Narudžbine*

U dnu stranice postoje tri dugmeta: “Pošalji podsetnik” – otvara se modal u kome će biti izlistane mejl adrese ljudi koji još uvek nisu popunili hranu za narednu nedelju i kojima će klikom na dugme “Pošalji” biti poslat mejl podsestnik ([slika 2](#_bookmark46)4), “Sačuvaj PDF” – generisaće se PDF sa datim podacima ([slika 2](#_bookmark45)5), “Onemogući/Omogući izbor jela” – svi korisnici dobijaju ili gube mogućnost izbora jela.



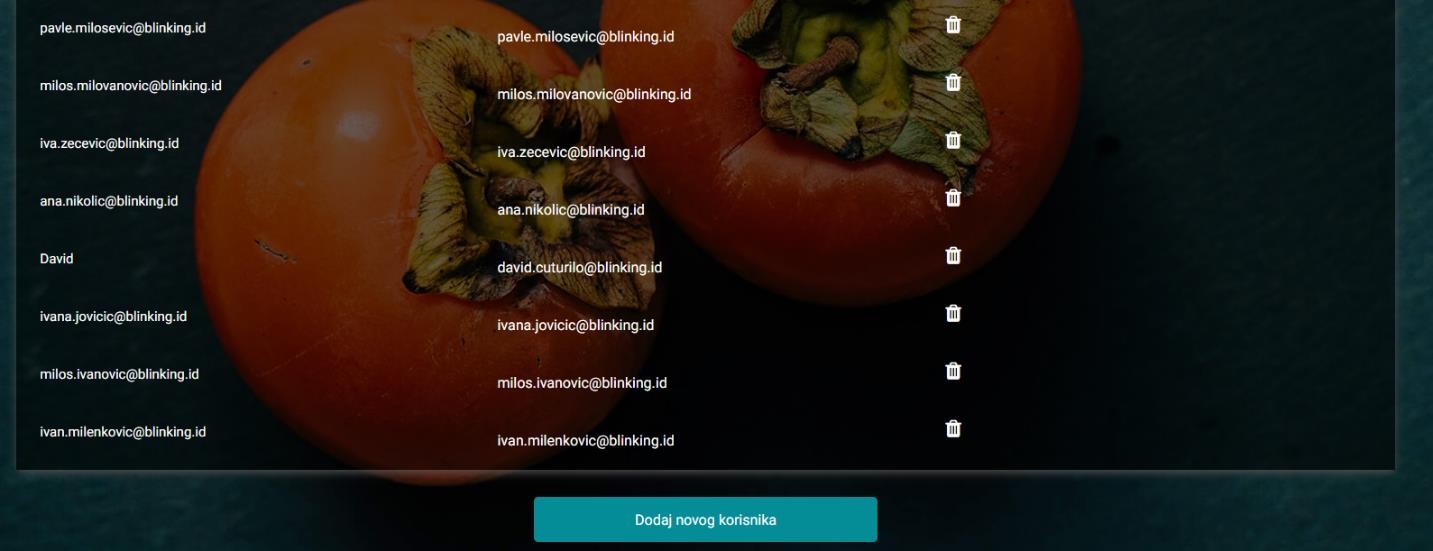
*Slika 24 Podsetnik*



*Slika 25 Narudzbine PDF*

### Svi korisnici

Stranici “Svi korisnici” može pristupati samo administrator, na njoj se nalazi prikazana tabela svih korisnika koji se klikom na ikonicu kantice mogu obrisati ([slika 2](#_bookmark50)6). Na dnu stranice se nalazi dugme “Dodaj korisnika”, klikom na dato dugme otvara se modalni prozor gde se može registrovati novi korisnik.



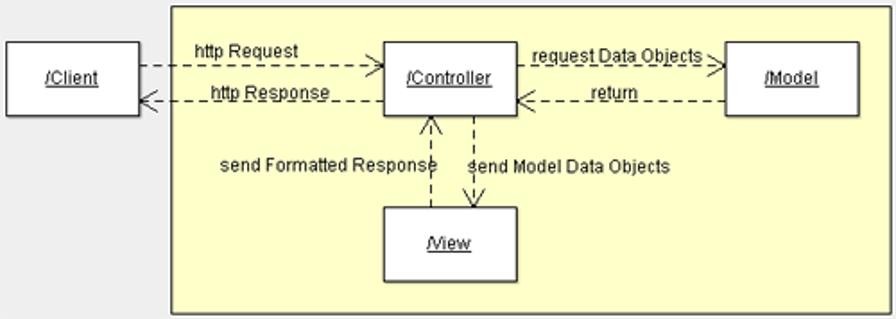
*Slika 26 Svi korisnici*

# Implementacija sistema

U prvom zaglavlju biće opisana arhitektura sistema, nakon toga imajući u vidu obim aplikacije u ovoj glavi će biti prikazani i objašnjeni samo najinteresantniji delovi koda.

### Arhitektura sistema

Za aplikaciju će biti korišćen *Model-View-Controller* projektni uzorak ([slika 27](#_bookmark54)). *Model* (entiteti aplikacije) predstavlja podatke i logiku aplikacije, on čuva i vraća entitete korišćene od strane aplikacije, obično iz baze podataka, i sadrži logiku aplikacije. *View* (klijentski deo) je odgovoran za formatiranje podatake, u *HTML* formatu, dobijene iz modela u obliku koji je pristupačan korisniku. Podaci mogu doći u različitim formatima od modela: jednostavni objekti, xml strukture, JSON. *Controller* (serverski deo) kada primi zahtev od klijenta odlučuje koji podaci se zahtevaju od modela onda pokreće model za obavljanje traženih poslova i šalje podatke na odgovarajući prikaz preslđujući mu podatke koji dolate iz modela.

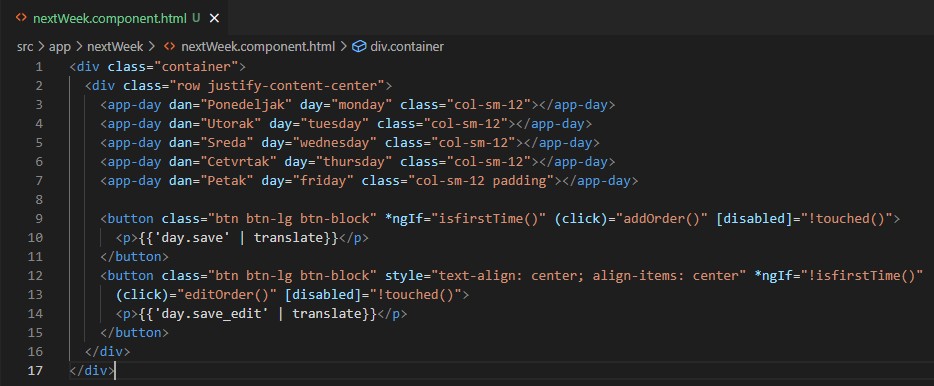


*Slika 27 Model-View-Controller*

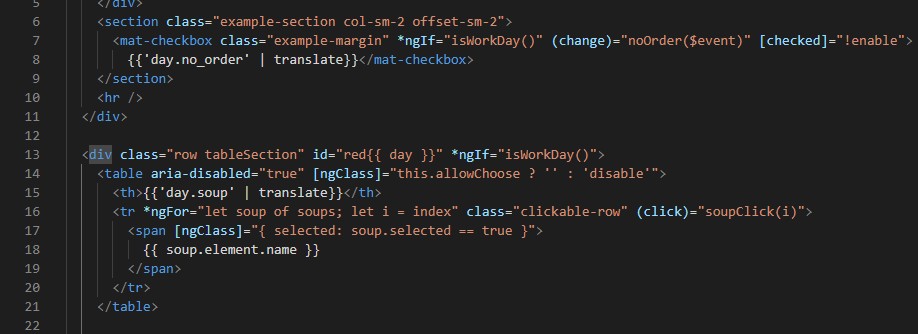
Baza podataka se sastoji od nekoliko tabela. Korisnički podaci se čuvaju u tabeli korisnik, koja sadrži sledeće tabele: email, korisničko ime, ime, šifra, sliku i ulogu. Predlozi su predstavljeni tabelom predlozi koja pored naslova i teksta predloga imaj i broj podrške tom predlogu. Jela su predstavljena tabelom koja sadrži ime i vrstu tog jela. Svaka naružbina korisnika je sačuvana u tabeli narudžbine gde se pored stranog ključa korisnika pamti i koja jela je on naručio i za koju nedelju. Nedeljna ponuda koju administrator sastavlja se čuva u istoimenoj tabeli i sadrži jela po danima koja su u ponudi.

### Početni ekran

Početna stranica se sastoji od pet komponenti(za svaki dan po jedna) i dva dugmeta koja će biti kombinovano prikazivana u zavisnosti od toga da li je nedeljna ponuda prvi put odabrana ili se samo izmenila što se određuje funkcijom isFirstTime() ([slika 2](#_bookmark58)8). Komponenta day na svom početku ima jedan check-box koje će korisnik čekirati u slučaju da dati dan ne dolazi na posao. Ispod toga se za svaku vrstu jela posebno nalazi po jedna tabela sa nazivima jela i klikom na određeno jelo korisnik ga bira, nakon čega će se dato jelo biti istaknutno na vrhu date tabele ([slika](#_bookmark57) [2](#_bookmark57)9).



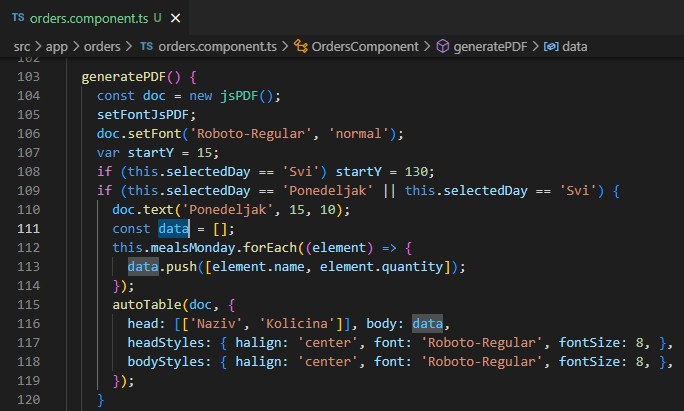
*Slika 28 Početni ekran*



*Slika 29 Komponenta day*

### PDF preuzimanje

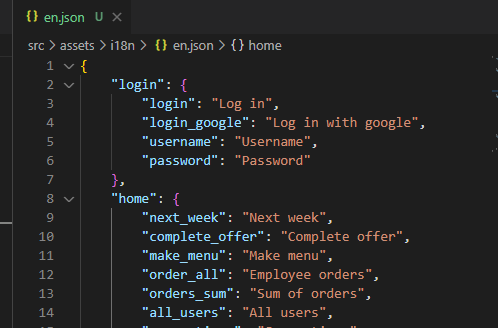
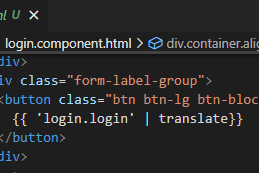
Prilikom preuzimanja tabele u PDF prvo moramo da nrapvimo objekat tipa jsPDF, podesimo font odredimo margine početka dokumenta. Pozivom metode autoTable formiramo tabelu kojoj moramo da zadamo nazive kolona (head) njena polja (body) i njihove formate (headStyles i bodyStyles) ([slika](#_bookmark60) 30).



*Slika 30 PDF preuzimanje*

### Dvojezičnost

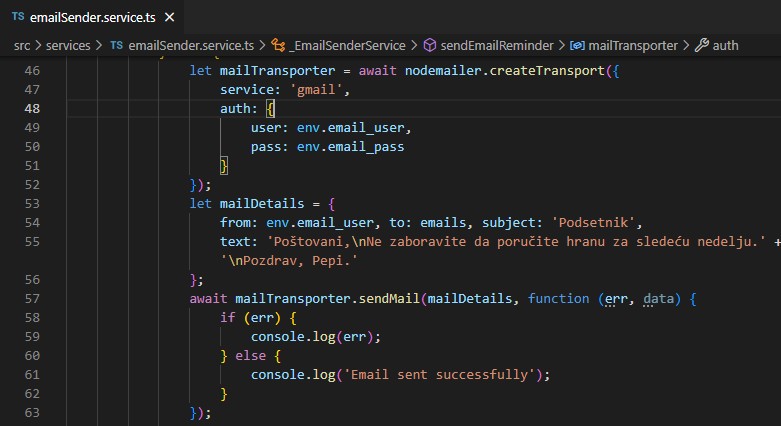
Dvojezičnost je ostvarena uz pomoc biblioteke *ngx-transalte*. Potrebno je imati za svaki od podržanih jezika po jedan json file u kome će se nalaziti prevodi za taj jezik ([slika 3](#_bookmark60)2). Zatim je dovoljno unutar HTML fajla samo se referencirati na želejenu poruku u json fajlu ([slika 3](#_bookmark60)1).



*Slika 31 Prevod Slika 32 JSON fajl sa prevodim*

### Slanje email-a

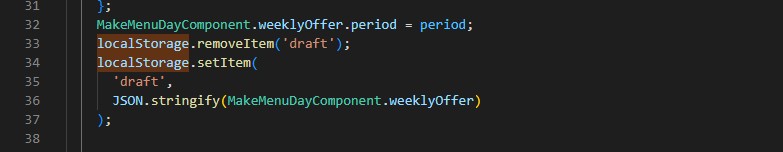
Slanje mejla se obavlja na serverskoj strani, pozivom metode createTransport nodemailer-a i setovanjem email-a, šifre mejla, teksta same poruke, nakon čega se poziva poziv sendMail ([slika](#_bookmark63) 33). Mejl se šalje prilikom prilikom registrovanja novog korisnika, sastavljanja nove nedeljne ponude ili kao podsetnik za one koji još nisu odabrali svoj jelovnik.



*Slika 33 Slanje email-a*

### Draft

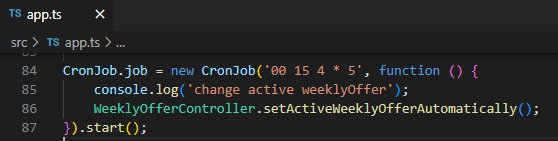
Prilikom sastavljanja jelovnika za sledeću nedelju često se dešava da je ketering firma napravila novo jela koje ne posotji u bazi podataka, i da administrator ne bi morao sva jela ispočetka da unosi, sva dotadašnja jela će biti sačuvana i učitana kad se ponovo vrati na datu stranicu ([slika 3](#_bookmark63)4).



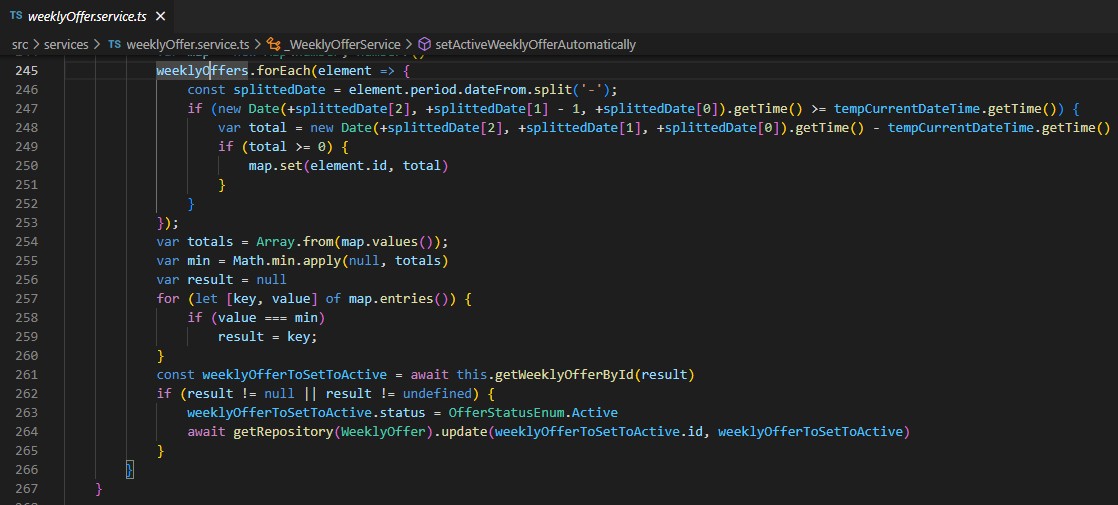
*Slika 34 Sačuvane izmene kod sastavljanja nedeljne ponude*

### Cron job

Aplikacija automatski svakog petka u 16:15 proverava i ako postoji nedeljna ponuda za sledeću nedelju setuje je na aktivnu. To se postiže preko *CronJoba* koji se okida u datom trenutku jednom nedeljno ([slika](#_bookmark66) 35).



*Slika 35 CronJob*

Metoda koja se poziva ima zadatak da setuje na aktivnu nedelju onu ponudu koja je najbliža u budućnosti, prvo se u mapu smeste sve one koje su u budučnosti a zatim se za svaku od njih odredi razlika u odnosu na sadašnji trenutka i za aktivnu nedelju se izabere ona sa najmanjom razlikom ([slika 3](#_bookmark66)6).

*Slika 36 Ažuriranje nedeljne ponude*

# Zaključak

Tema ovog diplomskog rada je izrada aplikacije za naručivanje hrane za potrebe jedne firme koja naručuje ketering za svoje zaposlene. Potreba za ovom aplikacijom se javila jer su dosadašnja rešenja imala dosta mana koja su realizacijom aplikacije u ovom radu otklonjena. Aplikacija za naručivanje hrane omogućuje korisniku prikaz ponude, brz i lak odabir hrane a administatoru uštedu vremena i lakše vođenje celog postupka oko naručivanja hrane.

Drugi deo rada je testiranje aplikacije, testiranje omogućuje proveru da li aplikacija ispravno radi. Aplikacija za naričivanje hrane je testirana pomoću dva testa: testiranje jedinice (Unit test), celovito testiranje (End-to-End test). Testiranje jedinice omogućuje da se testira samo jedan deo aplikacije, dok se celovitim testiranjem aplikacija testira od početka do kraja, odnosno kod celovitog testiranja je potrebno simulirati stvarnog korisnika u pretraživaču.

Testiranje jedinice je teklo paralelno sa razvojem projekta funkcionalnost po funkcionalnost ali i kad je projekat završen svaka stavka je detaljno testirana. Jedinično testiranje je otkrilo dosta propusta u aplikaciji koji su blagovremeno otklonjeni.

Celovito testiranje je sprovedeno na nivou firme koja broji 30 zaposlenih. Korisnicima je detaljno objašnjeno kako sama aplikacija funkcioniše kako bi mogli što bolje da istestiraju samu aplikaciju i prijave eventualne greške. I u ovoj fazi su otkrivene sitne greške i koje su u najkraćem roku otklonjene. Korisnici su osim grešaka predložili i manje izmene u dizajnu sajta od kojih su neke prihvaćene.

U toku same izrade projekta pojavljivali su se problemi koji su rešavani u toku izrade i koji nisu oduzeli previše vremena. Nakon odrađene aplikacije kao stavka koja je mogla biti drugačije urađena je instanciranje administratora u sistem, jedini način da se on doda je ručno slanje API zahteva, pa nekome ko nije stručan to bi moglo da predstavlja problem. Rešenje za to bi bila mogućnost dodavanja administratora iz same aplikacije preko korisničkog grafičkog interfejsa.

Eventualna poboljšanja aplikaciju mogu biti ta da administrator ne mora ručno sastavlja jelovnik već da samo učita dokument sa ponudom za sledeću nedelju koju dobija od ketering firme i da aplikacija sama sastavi nedelju ponudu. Takva funkcionalnost bi zahtevala strogo definisanu strukturu dokumenta ketering firme koju ona šalje svojim klijentima.

# Literatura

*Angular material.* n.d. https://material.angular.io/ (pristupano 9 2022).

*Angular.io.* n.d. https://angular.io/ (pristupano 9 2022).

*AngularJS.* 2022. https://sr.m.wikipedia.org/sr/AngularJS (pristupano 9 2022). Ayaz, Muhammad Ahsan. "Angular kuvar." In *Angular kuvar*, 55. Packt, 2021. *ExpressJS.* n.d. https://en.wikipedia.org/wiki/Express.js (pristupano 9 2022).

*ExpressJS official site.* n.d. https://expressjs.com/ (pristupano 9 2022).

*JavaScript.* 2022. https:/[/www](http://www.oxfordwebstudio.com/da-li-znate/sta-je-javascript.html).[oxfordwebstudio.com/da-li-znate/sta-je-javascript.html](http://www.oxfordwebstudio.com/da-li-znate/sta-je-javascript.html)

(pristupano 9 2022).

*Node.js.* 2022. https://sr.m.wikipedia.org/sr-ec/Node.js (pristupano 9 2022).

*Npm packages.* n.d. https:/[/www](http://www.npmjs.com/).[npmjs.com/](http://www.npmjs.com/) (pristupano 9 2022).

*Postgresql.* 2022. https://falcon-tech.rs/blog/postgresql-baze-podataka-obuka-instalacija/

(pristupano 9 2022).

*What is JavaScript?* n.d. https:/[/www](http://www.freecodecamp.org/news/what-is-javascript-definition-of-js/).[freecodecamp.org/news/what-is-javascript-definition-of-js/](http://www.freecodecamp.org/news/what-is-javascript-definition-of-js/) (pristupano 9 2022).

# Spisak

[Slika 1 Excel tabela kao jedno od postojećih rešenja 2](#_bookmark4)

Slika 2 Predlog novog rešenja 3

Slika 3 JavaScript 4

Slika 4 Angular 6

[Slika 5 Klijent server komunikacija 7](#_bookmark13)

[Slika 6 Angular komunikacija sa serverom 8](#_bookmark15)

Slika 7 ExpressJS 10

Slika 8 Node.JS 10

Slika 9 PostgreSQL 11

[Slika 10 Prijavljivanje 13](#_bookmark25)

Slika 11 Prijavljivanje preko Google-a 14

Slika 12 Registraciija korisnika 14

Slika 13 Promena lozinke I korisničkih podataka 15

Slika 14 Predlozi 15

Slika 15 Jelovnik 16

Slika 16 Brisanje jela 16

Slika 17 Ažuriranje jela 16

Slika 18 Dodavanje jela 17

Slika 19 Sastavljanje jelovnika 17

Slika 20 Promena jezika 18

[Slika 21 Porudzbine svih zaposlenih 19](#_bookmark40)

[Slika 22 Pdf prikaz porudzbina svih zaposlenih 19](#_bookmark42)

Slika 23 Narudžbine 20

[Slika 24 Podsetnik 21](#_bookmark47)

[Slika 25 Narudzbine PDF 21](#_bookmark48)

[Slika 26 Svi korisnici 22](#_bookmark51)

[Slika 27 Model-View-Controller 23](#_bookmark55)

Slika 28 Početni ekran 24

Slika 29 Komponenta day 24

Slika 30 PDF preuzimanje 25

Slika 31 Prevod 25

Slika 32 JSON fajl sa prevodim 25

Slika 33 Slanje email-a 26

Slika 34 Sačuvane izmene kod sastavljanja nedeljne ponude 26

Slika 35 CronJob 27

Slika 36 Ažuriranje nedeljne ponude 27