

KOMUNALNI ASISTENT

Sikora, Antonio

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Virovitica University of Applied Sciences / Veleučilište u Virovitici**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:165:448946>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-02-20**

Repository / Repozitorij:



[Virovitica University of Applied Sciences Repository - Virovitica University of Applied Sciences Academic Repository](#)



VELEUČILIŠTE U VIROVITICI

Preddiplomski stručni studij Računarstva

ANTONIO SIKORA

KOMUNALNI ASISTENT WEB APLIKACIJA
ZAVRŠNI RAD

VIROVITICA, 2021.

VELEUČILIŠTE U VIROVITICI

Preddiplomski stručni studij Računarstva

KOMUNALNI ASISTENT WEB APLIKACIJA
ZAVRŠNI RAD

Predmet: Projektiranje informacijskih sustava

Mentor:
dr. sc. Oliver Jukić, prof. v. š.

Student:
Antonio Sikora

VIROVITICA, 2021.



OBRAZAC 1b

ZADATAK ZAVRŠNOG RADA

Student/ica: SIKORA ANTONIO **JMBAG:** 0307015692

Imenovani mentor: dr. sc. Oliver Jukić, prof. v. š.

Imenovani komentor: -

Naslov rada:

Komunalni asistent – reporting wizard

Puni tekst zadatka završnog rada:

Sustav *Komunalni asistent* ima zadaću pomoći građanima u evidentiranju komunalnih problema na području grada. Jedan dio sustava je aplikacija kojom građanin može prijaviti komunalni problem bilo koje vrste, pri čemu se prijavom u sustav prosljeđuje kratki opis problema, lokacija problema (adresa, latituda i longituda) te opcionalno dodatni opis koji može biti tekst i/ili više fotografija. Nakon što je građanin prijavio problem, unutar sustava koristimo terminologiju da je građanin otvorio *ticket* (zapis).

Zadatak Vašeg završnog rada je napraviti web aplikaciju koja služi za izradu izvješća. Aplikacija mora imati mogućnost pregleda *ticketa* u tabelarnom obliku, uz filtriranje po statusu i vremenu otvaranja (odabir iz kalendara). Potrebno je omogućiti i filtriranje po geografskoj pripadnosti, na način da se u skočnom prozoru odabere pravokutno područje a da se u tablici prikazuju samo *ticketi* koji pripadaju odabranom području.

Odabirom *ticketa* ispod tablice ili u skočnom prozoru se pojavljuju detalji o njemu, koji obavezno moraju sadržavati i unesene fotografije.

Na osnovu podataka iz baze, potrebno je omogućiti najmanje ova izvješća:

- Pregled broja *ticketa* za vremenski period grupiranih po statusima
- Geografski pregled otvorenih *ticketa* za vremenski period (*heat map*)
- Prikaz broja trenutno otvorenih *ticketa* koji su stariji od N dana.

Sustav naslanja se na centralnu bazu podataka u kojoj su evidentirani *ticketi*.

U sklopu završnog rada morate kreirati bazu podataka na platformi MSSQL server, koja će biti osnova za izradu aplikacije. Potrebno je kreirati osnovne tablice, za pojedina polja koja imaju predefimirane vrijednosti potrebno je kreirati i šifranike. U osnovnim tablicama potrebno je definirati primarne ključeve i indekse. Bazu je potrebno napuniti testnim

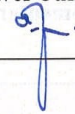
podacima. Nakon kreiranja baze podataka potrebno je nekim od formalnih jezika unutar jezičnog prostora napraviti specifikaciju aplikacije koja će biti zatim implementirana kroz završni rad. Za pristup aplikacije podacima u bazi potrebno je realizirati sučelje koje može biti korišteno i za druge svrhe, u bilo kojoj od tehnologija.

Datum uručenja zadatka studentu/ici: 28.07.2021.

Rok za predaju gotovog rada: 06.09.2021.

Mentor:

dr. sc. Oliver Jukić, prof. v. š.



Dostaviti:

1. Studentu/ici
2. Povjerenstvu za završni rad - tajniku

KOMUNALNI ASISTENT

REPORTING WIZARD

SAŽETAK - Cilj rada jest osmisliti i napraviti web aplikaciju u okviru sustava Komunalni asistent koji ima zadaću pomoći djelatnicima. Aplikacija se sastoji od tri dijela. U prvom dijelu moguće je pregledati sve tickete po statusu, radniku koji ga je ažurirao, rasponu datuma ili starosti. Također je moguće odabrati geografsko područje na mapi i prikazati samo tickete koji se nalaze unutar tog područja. Drugi dio aplikacije prikazuje Google mapu s heat elementom. Prikazuju se ticketi s krugom u različitim bojama. Crvena boja u krugu znači da se na tome području nalazi više ticketa. Preko ovog dijela aplikacije na jednostavan se način može saznati koje područje ima najviše prijavljenih problema. Treći dio aplikacije sastoji se od prikaza ticketa po radniku. Odabirom radnika prikazu se ticketi koje je taj radnik ažurirao te postotak riješenih i odbačenih ticketa. U radu će biti opisane korištene tehnologije, arhitektura aplikacije te rad i funkcionalnosti web aplikacije. Aplikacija koristi MySQL bazu podataka, Angular na klijentskoj strani i Node.js na serverskoj strani.

Ključne riječi: web aplikacija, ticket, baza podataka, Angular, Node.js

SUMMARY - *This paper aims to design and create a web application within the system of Reporting wizard, which has the task of helping the employees. The application consists of three parts. In the first part, it is possible to view all the tickets by status, the worker who updated them, the date range, or the age of the ticket. It is also possible to select a geographical area on the map and display only the tickets that are within the given area. The second part of the application shows a Google map with a “heat element”. Tickets in an area are displayed in different colors. The red color in the area means that there are more tickets in that area. Through this part of the application, you can easily find out which area has the most reported problems. The third part of the application consists of displaying tickets per employee. When an employee is selected, the tickets that the employee has updated are displayed, as well as the percentage of resolved and rejected tickets. The paper will describe the technologies used, the architecture of the application, and the operation and functionality of the web application. The application uses MySQL database, Angular on the client-side and Node.js on the server-side.*

Keywords: *web application, ticket, database, Angular, Node.js*

SADRŽAJ

1. UVOD	1
2. KORIŠTENE TEHNOLOGIJE	2
2.1. ANGULAR	2
2.1.1. HTML	3
2.1.2. CSS	4
2.1.3. TypeScript	5
2.1.4. Bootstrap	6
2.1.5. jQuery	6
2.2. Node.js	7
2.2.1. JavaScript	8
2.2.2. Express.js	8
2.3. MS SQL Server	9
2.3.1. SQL jezik	9
3. ARHITEKTURA APLIKACIJE	11
3.1. Klijenti	12
3.2. Server	12
3.3. API	12
3.4. Odgovor	12
3.5. Baza podataka	13
4. FUNKCIONALNOSTI APLIKACIJE	14
4.1. Login	14
4.2. Početna stranica	15
4.3. Heat mapa	18
4.4. Navigacija	22
4.5. Radnici	22
5. ZAKLJUČAK	25
6. LITERATURA	26
7. POPIS ILUSTRACIJA	27

1. UVOD

Web aplikacije su web stranice koje rade na principu povezanosti klijent-server i smatraju se računalnim programima koji se izvode u internet pregledniku. Klijent-server komunikacija znači da server upravlja zahtjevima i upitima samog koda aplikacije. Klijent u svome web pregledniku zatraži izvođenje nekog dijela koda koji se šalje serveru na obradu i rezultat vraća korisniku na web preglednik. Većinom, web aplikacije koriste i neku bazu iz koje server dohvaća potrebne podatke.

Komunalni asistent je web aplikacija koja se bavi evidentiranjem i rješavanjem komunalnih problema. Ukoliko građani uoče neki komunalni problem mogu na jednostavan način to prijaviti nadležnima. Tako prijavljen problem u sustavu zovemo ticket. Ticket može imati tri stanja, a to su “otvoren”, “riješeno” i “odbačeno”. Za pregled ticketa koristimo se web stranicom za izradu izvještaja u kojoj tickete možemo filtrirati i vidjeti im točnu lokaciju na mapi. Web stranica za izradu izvještaja rađena je u Angularu, dok je server strana rađena u Node.js. Za bazu podataka korišten je MS SQL Server.

Na početku rada opisane su korištene tehnologije te su dani primjeri koda. U radu će biti opisan Angular u sklopu kojeg ćemo se dotaknuti HTML-a, CSS-a i TypeScript-a te Node.js u sklopu kojeg ćemo se dotaknuti JavaScript-a i Express.js kao najpotrebnijeg modula za ovaj sustav. U drugom dijelu rada detaljno ćemo opisati rad aplikacije i funkcije koje su korištene.

2. KORIŠTENE TEHNOLOGIJE

U ovome poglavlju objasniti ćemo tehnologije koje su korištene u izradi web aplikacije. Neke od tehnologija koje će biti opisane jesu HTML, CSS, JavaScript, TypeScript i MS SQL Server.

2.1. ANGULAR

Angular je okvir za razvoj aplikacija razvijen od strane Google-a 2009. godine. U prvoj verziji nazivao se AngularJS zato što se koristio JavaScript, a od druge verzije naziva se samo Angular. Druga verzija pojavljuje se 2016. godine. U drugoj verziji više se ne koristi JavaScript, nego TypeScript. Svrha TypeScript-a jest nadomjestiti nedostatke JavaScript-a. Osnovna namjena Angular okvira je razvoj jednostraničnih¹ web aplikacija.

Angular aplikacija sastoji se od skupa komponenti. Skup komponenti tvori logičku cjelinu, koju onda grupiramo u modul. Svaka Angular aplikacija mora se sastojati od barem jednog modula, a taj modul treba sadržavati korijensku komponentu. Sve ostale komponente ugnježđuju se unutar korijenske komponente. Same komponente unutar Angular aplikacije pojavljuju se u obliku TypeScript klase. Unutar te klase dodajemo aplikacijsku logiku naše komponente te deklariramo i definiramo članove koji se trebaju nalaziti unutar komponente. Pošto je komponenta definirana klasom, više instanci te komponente može se pojaviti u različitim dijelovima aplikacije, a time se postiže ponovna upotrebljivost komponente. Unutar te komponente moguće je ukomponirati i druge komponente. Vizualni dio komponente naziva se prikaz, a njega definira predložak. Predložak se sastoji od HTML koda i koda specifičnog za Angular aplikaciju. Pošto koristimo HTML, moguće je dodati i CSS kod kojim uljepšavamo vizualni prikaz.

Angular aplikacija pokreće se u terminalu preko komande “ng serve” ukoliko smo u pravom direktoriju. Tada se pokreće compiler od Angular-a koji pretvara TypeScript kod u JavaScript kod i javlja greške na koje naleti. Komanda “ng serve” služi za testiranje aplikacije dok se radi na njenom kodu. Ukoliko želimo pripremiti aplikaciju za produkciju potrebno je upisati “ng build” komandu s kojom dobijemo normalnu HTML i JavaScript stranicu koju možemo prenijeti na bilo koji hosting² za web stranice.

¹ jednostranična – znači da postoji jedan HTML dokument unutar kojeg stavljamo sve druge HTML elemente. Sve stranice se prikazuju unutar index.html.

² hosting – server na koji možemo prenijeti našu web aplikaciju i pristupiti joj preko web preglednika

Jedna od istaknutijih funkcionalnosti Angular-a je ta što se prikaz varijabli mijenja real time. Real time znači da ukoliko negdje u kodu kasnije promijenimo vrijednost varijable, ona će se prikazati s novom vrijednosti istog trenutka bez potrebe za osvježavanjem stranice od strane korisnika.

2.1.1. HTML

HTML (*eng. HyperText Markup Language*) je prezentacijski jezik koji se koristi za izradu stranica. Služi za označavanje hipertekstualnih dokumenata, odnosno stvaranje hiperveza između HTML dokumenata. Bitno za napomenuti je kako HTML nije programski jezik, nego opisni. Opisni znači da sadrži skup uputa koje upućuju web preglednike kako prikazati sadržaj opisan u HTML dokumentu.

HTML dokument prepoznaje datoteke “html” ili “htm” po ekstenziji. Moguće ga je stvoriti u bilo kojem uređivaču teksta. Upute web pregledniku kako prikazati sadržaj stranice pišemo u oznakama ili tzv. tag-ovima. Oznake se pišu unutar znakova “<” i “>”. Oznake se mogu proširiti atributima koji se pišu unutar šiljastih zagrada nakon naziva oznake i služe za dodavanje informacija o elementu. Većina oznaka ima početnu i završnu oznaku. Završne oznake unutar šiljastih zagrada ispred naziva same oznake imaju kosu crtu i po tome ih prepoznajemo.

Neki od uređivača teksta koji se mogu koristiti jesu: Notepad, Visual Studio Code, Notepad++, Sublime text, a neki od češće korištenih web preglednika su Mozilla Firefox, Opera, Internet Explorer i Google Chrome.

Slika 1. Osnovni HTML

```
<html>
  <head>
    <title>Naslov dokumenta</title>
  </head>
  <body>
    Tijelo dokumenta
  </body>
</html>
```

Izvor: autor

2.1.2. CSS

CSS je opisni jezik koji služi za oblikovanje i dizajniranje stranice kako bi korisnik imao što bolje korisničko iskustvo te kako bi stranica bila pregledna. CSS se isto kao i HTML može uređivati u nekom od uređivača teksta, ali mu je ekstenzija datoteke “css”. Uključuje se u zaglavlje HTML dokumenta oznakom “<link>”. Pokraj uključivanja u zaglavlje moguće ga je pisati i linijski u HTML dokumentu tako da svakom elementu dodamo željeni stil ili možemo pisati stilove odvojeno unutar “<style>” elementa. Preporučeni način pisanja CSS-a je unutar posebne datoteke pošto na ostala dva načina lako dolazi do zalihosti pa se isti mogu ponavljati više puta za isti element.

Sintaksa CSS-a je jednostavna. Kako bi se promijenilo neko vizualno svojstvo nekog HTML elementa, unutar CSS datoteke navodi se CSS selektor iza kojega idu vitičaste zagrade unutar kojih se navode parovi CSS svojstava i njihovih vrijednosti. Svojstvo i vrijednost odvajaju se dvotočkom, a parovi unutar vitičastih zagrada odvajaju se znakom “;”. Selektorom odabiremo kojem HTML elementu ili skupu elemenata želimo dodijeliti određeno svojstvo i vrijednost. Željenom HTML elementu možemo pristupiti preko vrijednosti atributa klase, vrijednosti atributa ID ili preko naziva HTML elementa. Vrijednosti atributa klase pristupamo znakom “.”, vrijednosti atributa ID pristupamo znakom “#”, a nazivu HTML elementa pristupamo upisivanjem naziva elementa bez ikakvih drugih znakova.

Slika 2. Primjer CSS koda

```
#info-ikonica{
  font-size: 30px;
  cursor: pointer;
}
```

Izvor: autor

Pomoću CSS-a moguće je tekstu mijenjati font, veličinu fonta, boju, precizno pozicionirati na određenom dijelu stranice ili dodavati animacije. U primjeru sa slike, HTML elementu s vrijednosti atributa ID “info-ikonica” postavili smo veličinu fonta na 30px te promijenili izgled miša na “pointer” prelaskom preko elementa.

2.1.3. TypeScript

TypeScript je proširenje JavaScript-a koje je razvio Microsoft. Ima sve značajke JavaScript-a. Kod pisan u TypeScript-u prvo se prevodi u JavaScript te se nakon toga izvršava u web pregledniku. TypeScript je zapravo samo proširenje JavaScript-a, tako da su nam u njemu dostupne sve funkcionalnosti koje su dostupne i u JavaScript-u. TypeScript koristi ekstenziju datoteke “ts”. Promjena ekstenzije datoteke iz “js” u “ts” neće utjecati na izvršavanje koda. Prevođenje koda s TypeScript prevoditeljem omogućuje nam da greške pronađemo već tijekom prevođenja.

Slika 3. Primjer definiranja tipa varijable

```
mapa: any;|
zoom: number = 7;
```

Izvor: autor

Jedna od odlika TypeScript-a je definiranje tipova varijabli. U samom JavaScript-u nema definiranja varijabli, dok TypeScript dodaje mogućnost da možemo definirati varijablu kao number, string, boolean ili neki drugi tip. Tip vrijednosti definiramo na način da iza imena varijable pišemo znak “:” i tip. U primjeru sa slike definirali smo varijablu mapa kao bilo koji tip, a varijablu zoom kao tip number. Varijabli mapa možemo dodijeliti bilo koju vrijednost, a varijabli zoom možemo dodijeliti samo brojeve.

Slika 4. Primjer definiranja tipa funkcije

```
function fZbroji(a: number, b: number): number {
  return a + b;
}
```

Izvor: autor

Funkcijama također možemo odrediti tip podataka koje one vraćaju. U primjeru sa slike definirali smo tip funkcije kao number. Ukoliko bismo funkciji pokušali proslijediti vrijednost koja nije number ili ako bismo njezinu vrijednost pokušali pridružiti varijabli čiji tip nije number ili any, prevoditelj bi nam javio grešku. Također bi javio grešku i ako bismo pokušali pozvati funkciju bez parametara. U TypeScript-u su svi parametri obavezni, dok su u JavaScript-u svi parametri opcionalni. Ukoliko želimo imati i opcionalne parametre, iza naziva varijable moramo staviti znak “?”.

TypeScript je objektno orijentiran programski jezik. Osim klasa, dostupna su nam i sučelja te apstraktne klase.

2.1.4. Bootstrap

Bootstrap je jedan od popularnijih HTML, CSS i JavaScript programskih okvira koji ubrzava razvoj responzivnih³ stranica. Besplatan je i može se lako dodati u bilo koji projekt. CSS i JavaScript datoteke od Bootstrap-a se dodaju na isti način kao i sve druge CSS i JavaScript datoteke.

Bootstrap je veliki skup HTML, JavaScript i CSS koda koji se primjenjuje na HTML elemente stranice. Da bismo koristili neku funkcionalnost Bootstrap-a potrebno je u atribut “class” HTML elementa dodati jedan od naziva Bootstrap klasa. Znači više linija u HTML-u, CSS-u i JavaScript-u zamjenimo s jednom riječju.

Dodavanje Bootstrap-a u Angular projekt odrađuje se na malo drugačiji način nego u obični HTML. Za korištenje Bootstrap-a u Angular-u potrebno je skinuti “ng-bootstrap” pomoću NPM-a⁴.

2.1.5. jQuery

jQuery je jedan od najčešće korištenih dodataka koje nudi JavaScript. Uz manje koda olakšava upotrebu JavaScript-a i omogućuje pisanje moćnih web aplikacija. Jedna linija koda mijenja više linija JavaScript koda. Podržan je od strane svih web preglednika. Osnovna sintaksa jQuery-ja je `$(selektor).akcija()`;

Slika 5. Primjer jQuery koda

```
Otvori(){
  if($("#div").attr('prikazan') == "false" ) {
    $("#div").css('display', 'table-row');
    $("#div").attr('prikazan', 'true');
  }else{
    $("#div").css('display', 'none');
    $("#div").attr('prikazan', 'false');
  }
}
```

Izvor: autor

³ responzivno – znači da stranica dobro izgleda i radi na svim vrstama i veličinama ekrana

⁴ NPM – punim nazivom Node Package Manager, je stranica s dodatcima za Angular i Node.js aplikacije

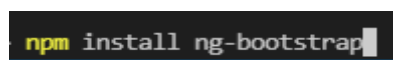
Na slici broj 5 može se vidjeti primjer jQuery koda. Ovo je funkcija koja provjerava je li atribut “prikazan” postavljen na “false”. Ukoliko je postavljen na “false” mijenjamo CSS atribut “display” elementu s ID-om “div” u “table-row” i postavljamo atribut “prikazan” na “true”. Ako ipak atribut “prikazan” nije “false” postavljamo CSS atribut “display” na “none” i atribut “prikazan” na “false”. U prvom dijelu prikazujemo skriveni div element, a u drugom dijelu ga skrivamo. HTML elemente možemo mjenjati po ID-u i nazivu klase. Ukoliko želimo promijeniti element s ID-om iza znaka “\$” u zagrade pišemo znak “#” i željeni ID. Ako želimo promijeniti element s klasom u zagrade pišemo znak “.” i naziv klase.

2.2. Node.js

Node.js je JavaScript okruženje koje koristi Chrome-ov V8 engine za prevođenje JavaScript koda u strojni kod. Nastao je 2009. godine, a dvije godine kasnije postao je open-source⁵ projekt. Glavni cilj Node.js-a jest pružiti siguran i lak način za izradu mrežnih aplikacija visokih performansi u JavaScript-u. Pošto se JavaScript tradicionalno izvršava u web pregledniku, pojavila se potreba za pokretanjem na poslužiteljima te se Node.js pokazao kao najpraktičnije rješenje. Razlika između Node.js-a i nekih drugih sličnih alata je što Node.js koristi jednu nit, dok većina drugih dostupnih alata koristi više niti. Node.js bazira se na događajima. Naredbe od korisnika dolaze do poslužitelja te ih on istovremeno i u realnom vremenu obrađuje unutar iste niti.

Bitno je napomenuti da kod Node.js dolazi s Node Package Managerom (NPM). Ideja NPM-a je javna dostupnost dodatnih komponenti koje su dostupne kroz jednostavnu instalaciju. Većinu tih dodatnih komponenti izradila je zajednica i podijelila s ostalima. Puno komponenti dodatno olakšava pisanje koda u Node.js-u. NPM možemo instalirati s grafičkim sučeljem ili kao dio komandnog sučelja. U komandnom sučelju da bismo instalirali dodatnu komponentu moramo upisati naredbu “npm install ime_komponente”. Ukoliko komponenta postoji, skinut će se i spremiti u mapu node_modules.

Slika 6. Komanda za instaliranje pomoću NPM-a



```
npm install ng-bootstrap
```

Izvor: autor

⁵ open-source – projekt otvorenog koda. Svi mogu vidjeti i uređivati kod projekta.

2.2.1. JavaScript

JavaScript je skriptni programski jezik interpretativnog tipa. Kod se direktno izvršava bez prevođenja u strojni jezik. Omogućuje implementaciju skripti koje će biti interaktivne prema korisniku i to iz njegovog vlastitog web preglednika. Podržavaju ga svi poznatiji web preglednici i nije potrebna instalacija ili konfiguracija kako bi ga koristili.

JavaScript datoteke koriste ekstenziju “js” i u HTML dokument se uključuju na dno “<body>” elementa oznakom “<script>”. JavaScript biblioteke uključujemo unutar “<head>” elementa.

Varijable u JavaScript-u deklariramo s riječju “var”. U JavaScript-u nema deklariranja varijabli po tipu, tako da varijabli koja je prvobitno imala spremljen broj možemo spremi tekst ili neki drugi tip podataka.

Pomoću JavaScript-a možemo manipulirati sadržajem web stranice. Možemo prikazivati ili skrivati elemente, mijenjati stil elemenata, reagirati na pritiske gumba i pročitati ili izmijeniti sadržaj.

2.2.2. Express.js

Express je najpopularniji Node web framework⁶. Dizajniran je za izradu web aplikacija i API-ja. Minimalistički je framework s puno opcija dostupnih kao dodatci (*eng. plugins*). Koristi ga dosta poznatih kompanija kao što su PayPal, Uber i IBM. API rute definiraju se na vrlo jednostavan način.

Slika 7. Primjer definiranja API rute

```
var tiketi      = require('./tiketi.js');
var statusi    = require('./statusi.js');

router.post('/api/tiketiPopis',      tiketi.tiketiPopis);
router.get('/api/statusiPopis',     statusi.statusiPopis);
```

Izvor: autor

U primjeru sa slike “router.post” znači da ćemo koristiti POST⁷ metodu poziva API-ja, a “router.get” znači da ćemo koristiti GET⁸ metodu. Prvi parametar znači da ćemo

⁶ framework – platforma koja pruža temelj za razvoj softverskih aplikacija

⁷ POST – način zahtjeva koji podržava HTTP. Služi za slanje podataka.

⁸ GET – način zahtjeva koji podržava HTTP. Služi za dohvaćanje podataka.

pozivati API “/api/tiketiPopis”, a drugi parametar označava da će se pozvati funkcija “tiketiPopis” koja se nalazi u dokumentu tiketi.js.

2.3. MS SQL Server

MS SQL Server je sustav za upravljanje relacijskim bazama podataka razvijen od strane Microsoft-a. Primarna funkcija mu je spremanje i zaprimanje podataka od drugih aplikacija. MS SQL Server i druga aplikacija mogu biti pokrenuti na istom računalu ili na različitim računalima s pristupom internetu. Pogodan je za velike poslovne sustave, ali i za male baze podataka. Koristi SQL jezik za upite nad bazom.

2.3.1. SQL jezik

SQL jezik je najpopularniji programski jezik za izradu, traženje, ažuriranje i brisanje podataka relacijskih baza podataka. Standardiziran je preko ANSI⁹ i ISO¹⁰ standarda. U relacijskim bazama podataka podaci se spremaju u tablice, a svrha SQL-a je dodati nove podatke u te tablice ili ih pronaći/ažurirati. Za programere je bitno znati CRUD¹¹ operacije, kao i kreiranje složenijih upita koji povezuju više tablica.

Slika 8. Primjer dohvaćanja podataka sa SQL-om

```
select ko_id, ko_korime from kasistent_korisnici where ko_korime='test' and ko_sifra='test'
```

Izvor: autor

U primjeru iz gornje slike možemo vidjeti način na koji se dohvaćaju podaci iz baze podataka. Dohvaćamo “ko_id” i “ko_korime” iz tablice “kasistent_korisnici” od korisnika koji ima “ko_korime” test i “ko_sifra” test. Isti takav upit koristi se za logiranje u aplikaciju.

Slika 9. Primjer spremanja podataka sa SQL-om

```
INSERT INTO kasistent_korisnici (ko_korime, ko_sifra, ko_ime, ko_prezime) VALUES ('pero', 'pero123', 'Pero', 'Perić')
```

Izvor: autor

U gornjem primjeru sa slike 6 prikazan je primjer spremanja podataka u bazu. U tablicu “kasistent_korisnici” upisujemo “ko_korime” pero, “ko_sifra” pero123, “ko_ime”

⁹ ANSI – American National Standards Institute je neprofitna organizacija koja se bavi razvojem i održavanjem volonterskih standarda za produkte, servise, sisteme i osoblje

¹⁰ ISO – Međunarodna organizacija za standardizaciju

¹¹ CRUD – kratica za create, read, update i delete (kreiranje, čitanje, ažuriranje i brisanje)

Pero i “ko_prezime” Perić. “ko_korime” je korisničko ime kojim će se korisnik prijaviti, “ko_sifra” je lozinka kojom će se korisnik prijaviti, a “ko_ime” i “ko_prezime” su ime i prezime korisnika.

Slika 10. Primjer brisanja podatka sa SQL-om

```
DELETE FROM kasistent_korisnici WHERE ko_id = 1
```

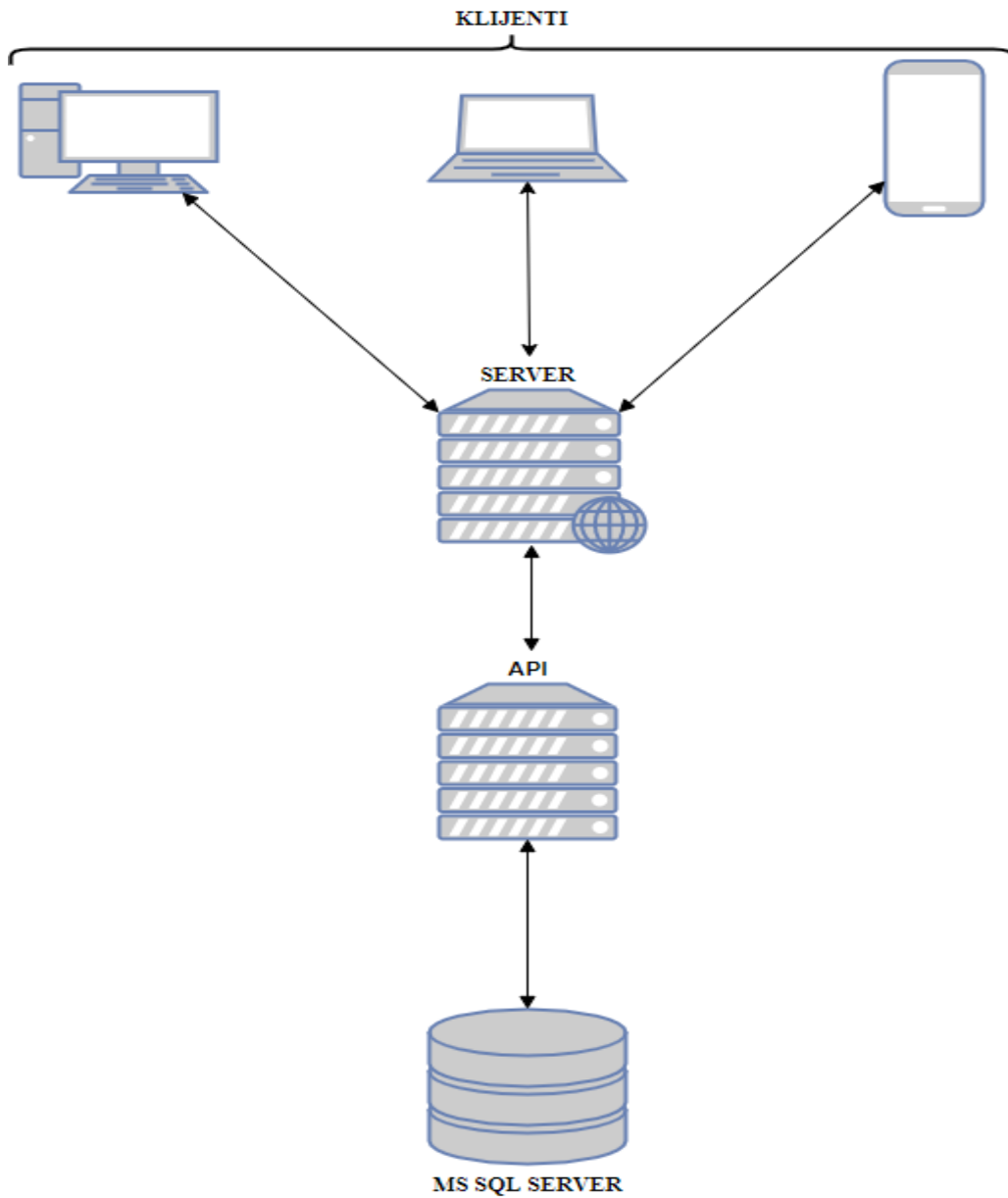
Izvor: autor

Na slici 7 može se vidjeti primjer koda za brisanje podataka iz baze. Iz tablice kasistent_korisnici brišemo korisnika koji ima “ko_id” 1. Ukoliko ne stavimo “WHERE” obrisat ćemo sve zapise koji se nalaze u tablici “kasistent_korisnici”.

3. ARHITEKTURA APLIKACIJE

U ovome dijelu rada opisana je arhitektura web aplikacije te put zahtjeva od klijenta do API-ja i nazad.

Slika 11. Arhitektura aplikacije



Izvor: autor

3.1. Klijenti

Klijenti su krajnji korisnici koji se spajaju sa svojih uređaja. Pod uređaje smatramo bilo koji uređaj koji ima pristup internetu i internet pregledniku. Većinom su to računalo, laptop, tablet ili mobitel. Klijenti su ti koji šalju zahtjev web serveru za prikaz nekog sadržaja u njihovom internet pregledniku.

3.2. Server

Server je zadužen za obradu zahtjeva koji dođu od klijenata. Ukoliko klijent želi prikazati određene tickete, poslat će zahtjev za tim ticketima serveru. Server će obraditi taj zahtjev i poslati ga API-ju.

3.3. API

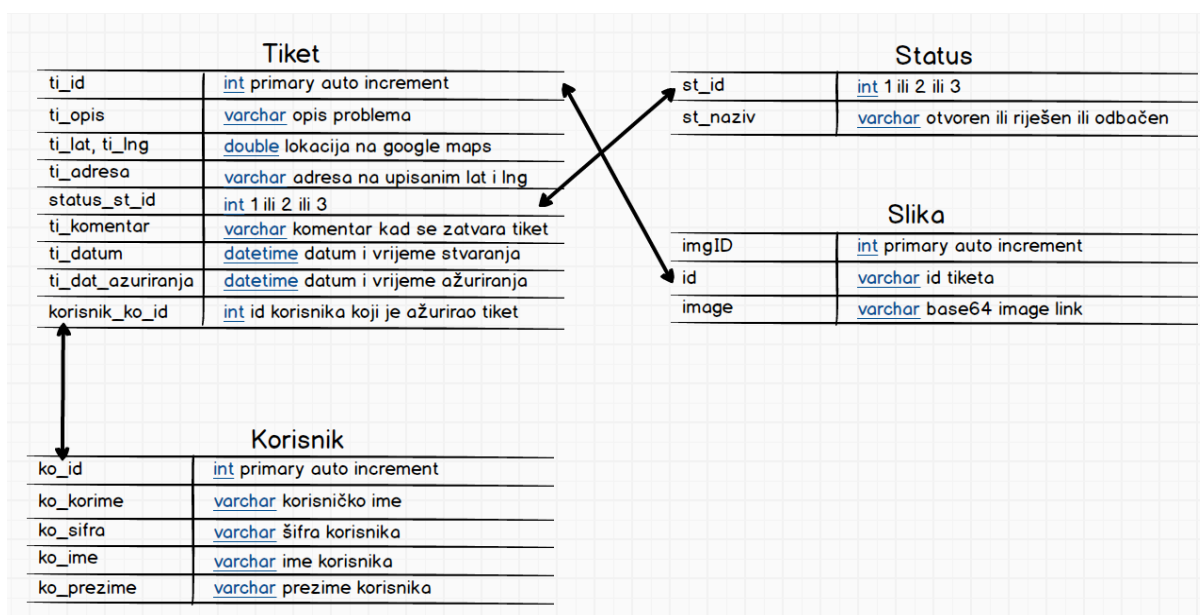
API je zadužen za pristup bazi podataka. Jedini ima pristup podacima koji se nalaze u bazi podataka. Ukoliko dobije zahtjev od servera za prikaz određenih ticketa, API će pristupiti bazi podataka, pronaći tražene tickete i poslati ih nazad serveru u JSON formatu.

3.4 Odgovor

Kada API pronađe tražene podatke i pošalje ih nazad serveru, server ih obradi i prikaže klijentima koji su slali zahtjev. Ukoliko je aplikacija dobro osmišljena i upiti prema bazi dobro napravljeni, ta komunikacija odvija se jako brzo.

3.5. Baza podataka

Slika 12. Struktura baze podataka



Izvor: autor

Baza podataka podijeljena je na četiri tablice. Tablica “Tiket” sastoji se od deset kolona. U nju se sprema svaki ticket koji korisnik napravi. U tablicu “Korisnik” spremaju se svi korisnici aplikacije. Sastoji se od pet kolona, od kojih su najbitnije “ko_korime” i “ko_sifra” radi logiranja u aplikaciju. Tablica “Status” je šifarnik statusa. U nju su upisana tri statusa. Prvi status je “riješen”, drugi status je “otvoren”, a treći status je “odbačen”. Zadnja tablica je “Slika”. U tu tablicu spremaju se slike koje korisnici prenesu.

Postoje tri međusobne veze među tablicama. Prva veza je veza između tablice “Tiket” i tablice “Korisnik”, a postoji zbog spremanja radnika koji je ažurirao ticket. Druga veza je veza između tablice “Tiket” i tablice “Status”. Postoji zbog dohvaćanja naziva statusa iz šifrnika. Treća veza je veza između tablice “Tiket” i tablice “Slika”. Ta veza postoji zbog povezivanja prenesenih slika s ticketom kojem pripadaju.

4. FUNKCIONALNOSTI APLIKACIJE

Aplikacija "Komunalni asistent" služi kao potpora komunalnim djelatnicima u poslu rješavanja komunalnih problema po gradu. Podijeljena je na tri web stranice. Prva web stranica osmišljena je za građane da prijavljuju komunalne probleme koje pronađu, druga je osmišljena za komunalne radnike da mogu rješavati prijavljene probleme, a treća je osmišljena za direktore komunalnih poduzeća. Direktori mogu u trećoj web stranici, koja se naziva "Izveštaji", pregledavati sve dosad prijavljene probleme i njihove statuse te kada su prijavljeni i riješeni. Također mogu pratiti radnike, odnosno koliko su ticketa obradili u određenom vremenskom periodu. Ako postoji potreba pregleda ticketa samo za određenu regiju ili za određeni kvart postoji mogućnost da preko Google mape označe tu regiju ili kvart. Moguće je pregledati i tickete po statusima na Google mapi s dodanim heat elementom. Dijelovi koji se na heat mapi prikazuju crveno jesu lokacije koje su često prijavljivane. Aplikacija je rađena u Angular-u i Node.js-u. U nastavku rada detaljno će biti opisano korištenje web aplikacije "Komunalni asistent izvještaji".

4.1. Login

Prva stranica s kojom se korisnik susreće jest "Login" stranica. Logirati se mogu sve osobe koje su unesene u bazu podataka. Ukoliko osoba ne unese točne login podatke dobiva poruku kako su korisničko ime ili lozinka netočni.

Slika 13. Login

The image shows a login form with a white background and a light gray border. At the top, the word "LOGIN" is centered in a bold, black, sans-serif font. Below it, there are two input fields. The first field is labeled "Korisničko ime" (Username) and the second is labeled "Šifra" (Password). Both labels are centered above their respective fields. Below the input fields, there is a prominent blue button with the text "LOGIRAJTE SE" (Log In) in white, uppercase letters.

Izvor: autor

4.2. Početna stranica

Slika 14. Početna stranica



The screenshot shows a web interface for ticket management. At the top, there are filter options: a dropdown for 'Izaberite korisnika', a dropdown for 'Stariji od: 0 dana', a date input 'Od: dd.mm.yyyy', a date input 'Do: 19.08.2021', and another dropdown for 'Izaberite status'. Below the filters is a table with the following data:

Broj tiketa	Status tiketa	Vrijeme otvaranja	Vrijeme ažuriranja	Operater	Komentar
1	Riješen	01.08.2021	13.08.2021	Antonio Sikora	
2	Otvoren	10.08.2021	-	-	-
3	Odbačen	14.08.2021	16.08.2021	Antonio Sikora	
4	Otvoren	19.08.2021	-	-	-
5	Otvoren	19.08.2021	-	-	-

Izvor: autor

Nakon logiranja korisnik će biti usmjeren na početnu stranicu aplikacije. Na početnoj stranici nalaze se svi ticketi koje je moguće filtrirati po starosti, datumu otvaranja, statusu ili operateru koji je ažurirao ticket. S lijeve strane iznad tablice nalazi se krug sa znakom "!". Pritiskom na tu ikonicu dobijamo modal koji prikazuje statistiku otvorenih, riješenih i odbačenih ticketa.

Slika 15. Modal statistike

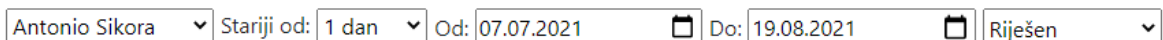


Izvor: autor

Desno od ikonice za statistiku nalazi se ikonica karte. Pritiskom na tu ikonicu otvara se modal s Google mapom. Taj modal služi za izbor prikaza ticketa unutar određene regije. Ukoliko pritisnemo lijevu tipku miša i povučemo po mapi nacrtat će se pravokutnik. Kada

se pravokutnik nacрта prikazat će se samo ticketi koji se nalaze unutar nacrtanog pravokutnika. Ukoliko želimo obrisati pravokutnik pritisnemo tipku resetiraj. Moguće je odabrati određenu regiju ili određeni kvart unutar grada. Nakon odabira područja u tablici će se prikazivati samo ticketi unutar tog područja. Pokraj izbora područja moguće je još točnije odrediti tickete koje želimo vidjeti preko filtera. Ukoliko pritisnemo na statistiku prikazivat će se samo brojevi trenutno prikazanih ticketa u tablici.

Slika 16. Filteri

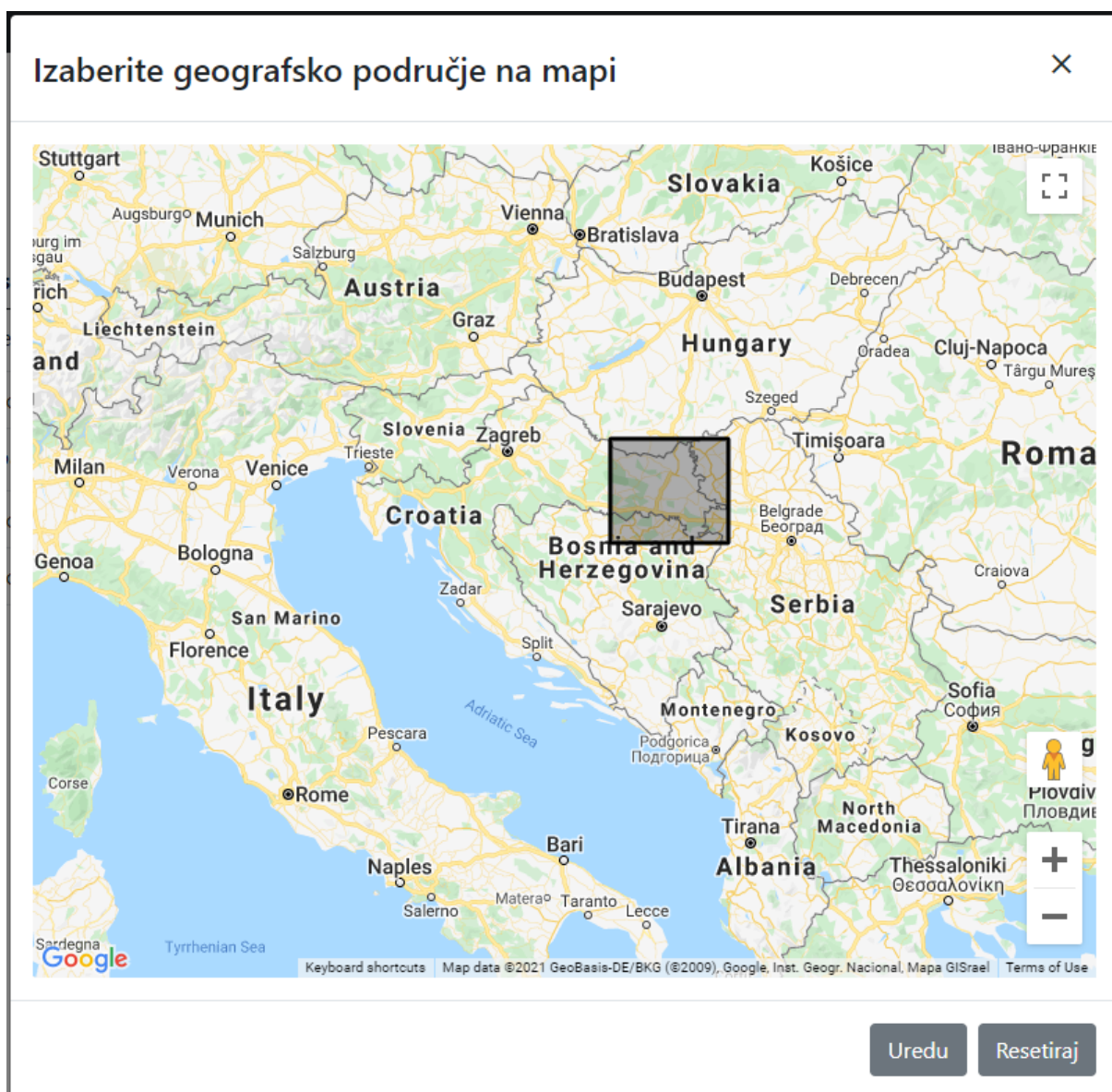


Antonio Sikora ▾ Stariji od: 1 dan ▾ Od: 07.07.2021 📅 Do: 19.08.2021 📅 Riješen ▾

Izvor: autor

Filtrirati se može po radniku koji je ažurirao ticket, po starosti ticketa (do deset dana starosti), po datumima i statusu ticketa. Ukoliko nema ticketa s odabranim filterima tablica će biti prazna.

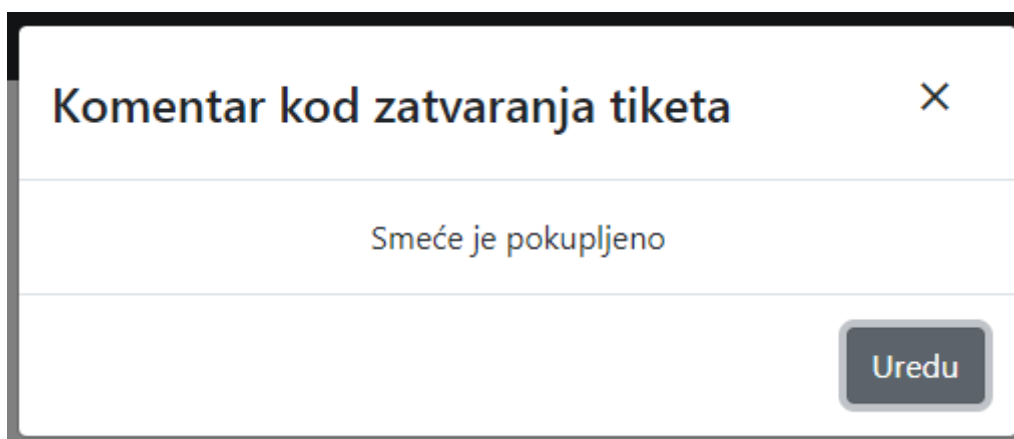
Slika 17. Modal za izbor geografskog područja



Izvor: autor

Ukoliko riješeni ili odbačeni ticket ima komentar, možemo ga pogledati pritiskom na ikonicu u stupcu “Komentar”. Pritiskom na ikonicu otvori se modal u kojem se nalazi napisani komentar od strane radnika komunalnog poduzeća. Za prikaz komentara izabrano je da se otvara modal zbog toga što bi komentari mogli biti i po nekoliko rečenica. Ovako stavljen komentar u modal neće poremetiti samu tablicu.



Slika 18. Modal s komentarom



Izvor: autor

Pritiskom bilo gdje na ticket otvara se dio s dodatnim informacijama o ticketu. Pod dodatnim informacijama nalaze se slike ukoliko ih je korisnik poslao. Ukoliko je više slika može ih se listati. Od drugih dodatnih informacija naveden je opis ticketa, latituda, longituda i adresa na tim koordinatama.

Slika 19. Detalji ticketa

1	Riješen	01.08.2021	13.08.2021	Antonio Sikora	
	Opis ticketa Poštovani, samo me zanima kada planirate pokupiti ovo smeće	Latituda 45.54	Longituda 18.05	Adresa Radnička 24 Đurđenovac	
2	Otvoren	10.08.2021	-	-	-
	Opis ticketa Netko je ovdje bacio jako puno smeća	Latituda 45.28	Longituda 18.80	Adresa Vukovarska 55	

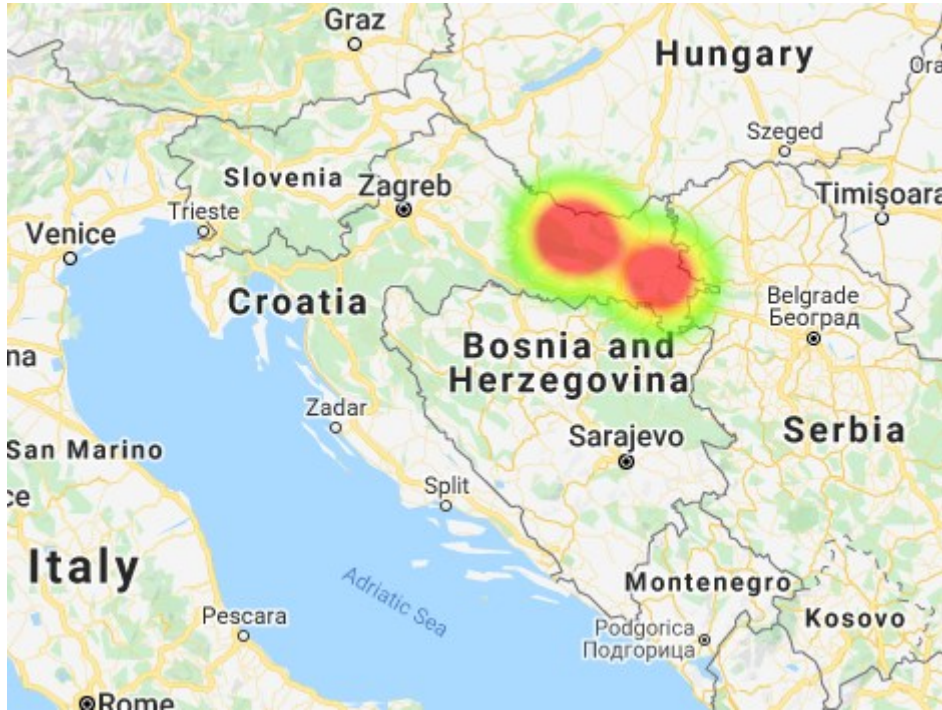
Izvor: autor

4.3. Heat mapa

Na stranici “Heat mapa” nalazi se Google mapa s dodanim heat elementom. Heat element je prikaz po bojama na mapi, koliko je ticketa prijavljeno na kojem području.

Ukoliko uvećamo mapu heat oznake se razdvajaju. Preko ove mape se na jednostavan način može saznati koje područje ima najviše problema s komunalnim otpadom.

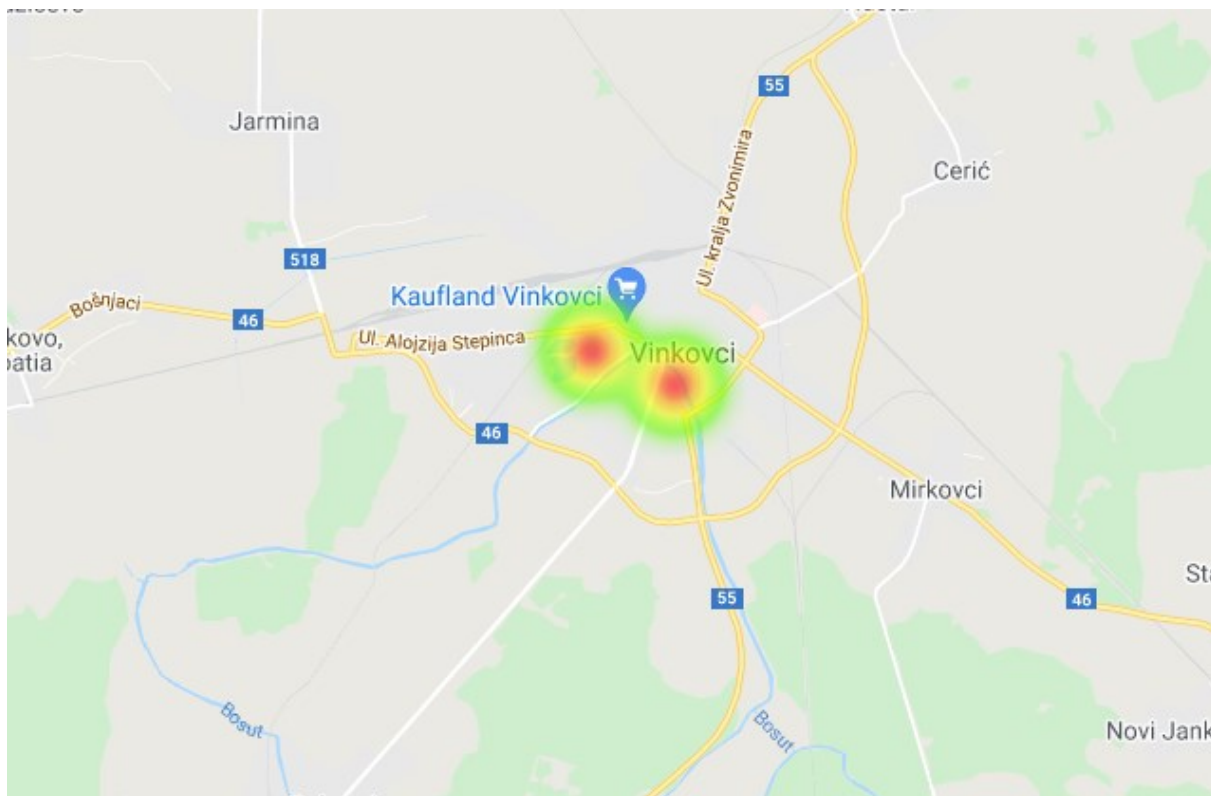
Slika 20. Heat mapa



Izvor: autor

Na slici 14 možemo vidjeti kako izgleda heat mapa kada područje nije uvećano. Po većem crvenom krugu možemo vidjeti da je na lijevom području otvarano više ticketa, negoli na desnom.

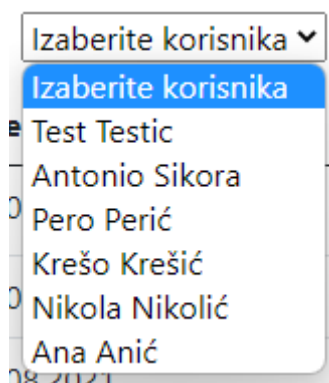
Slika 21. Heat mapa kad je uvećano određeno područje



Izvor: autor

Na gornjoj slici nakon uvećanja područja možemo vidjeti da se kreću prikazivati točnije lokacije svakog prijavljenog ticketa. Na trenutnom prikazu vidimo da su dva područja problematična po pitanju prijave građana.

Slika 22. Filter za izbor korisnika



Izvor: autor

Iznad heat mape nalaze se filteri. Prvi filter je filter za izbor radnika koji je ažurirao ticket. Ukoliko izaberemo korisnika “Antonio Sikora” prikazivat će se samo heat elementi

koje je taj radnik ažurirao. Ukoliko korisnik nije ažurirao niti jedan ticket dosad, mapa neće imati prikazanog heat elementa.

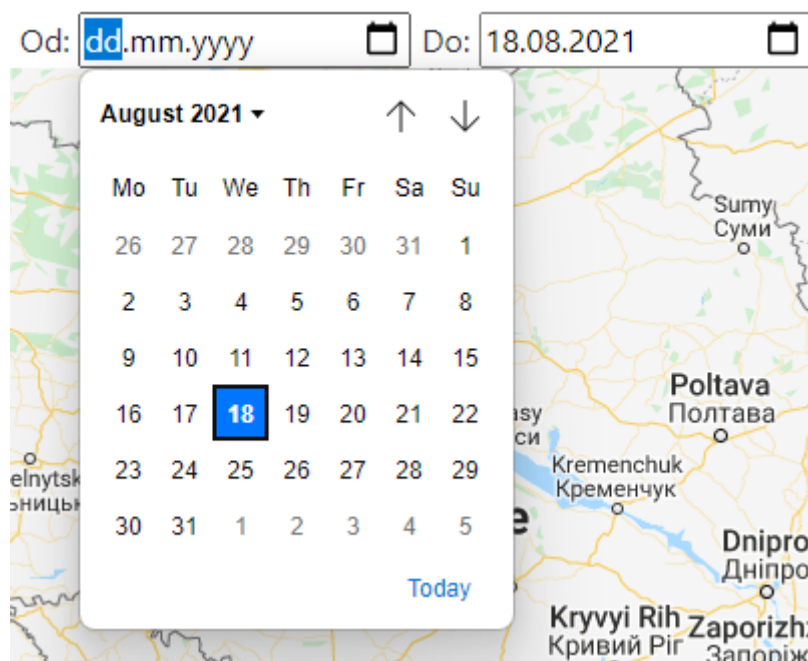
Slika 23. Filter za starost ticketa



Izvor: autor

Drugi po redu filter je filter za izbor starosti ticketa. Ovdje možemo odabrati prikaz ticketa po starosti. Ukoliko izaberemo “4 dana” prikazivat će se svi ticketi koji su stariji od četiri dana.

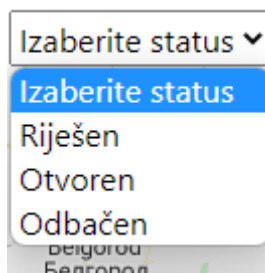
Slika 24. Filter po datumu



Izvor: autor

Treći po redu filter je filter za izbor raspona datuma otvaranja ticketa. Ukoliko izaberemo za “Od” datum “14.08.2021” i “Do” datum “18.08.2021” prikazivat će se samo ticketi koji su otvoreni unutar ta dva datuma.

Slika 25. Filter po statusu



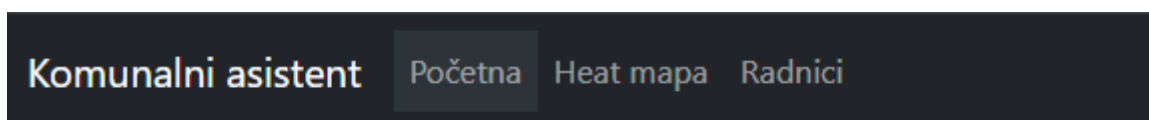
Izvor: autor

Zadnji filter s desne strane je filter za prikaz ticketa po statusu. Ukoliko izaberemo status “Riješen” prikazivat će se samo ticketi koji su riješeni.

4.4. Navigacija

Dizajn navigacije napravljen je uz pomoć Bootstrap-a. Prednost u Angular-u je ta što kada napravimo komponentu tu komponentu možemo postaviti u root¹² html i koristiti kroz sve druge komponente. To znači da smo postavljanjem koda “<app-navbar></app-navbar>” u root html dobili navigaciju na svim drugim dijelovima stranice.

Slika 26. Navigacija



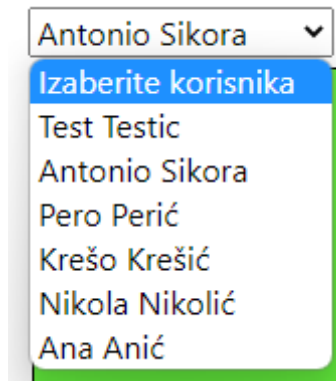
Izvor: autor

4.5. Radnici

Na stranici "Radnici" nalazi se pregled riješenih, odbačenih i ukupnih ticketa koje je određeni radnik ažurirao. Pokraj izbora radnika može se još izabrati i raspon datuma unutar kojih je rješavao tickete.

¹² root – glavni element Angular stranice

Slika 27. Izbor radnika



Izvor: autor

S lijeve strane nalazi se izbor korisnika. Pritiskom na "Izaberite korisnika" iz baze se dohvaćaju svi radnici. Kada se izabere radnik pojave se tri pravokutnika različitih boja. Ukoliko radnik nema ažuriranih ticketa unutar odabranog vremenskog perioda u pravokutnicima će pisati nula i tablice će biti prazne. Ako kod izbora korisnika odaberemo opciju "Izaberite korisnika" pravokutnici će nestati dok se ponovno ne odabere neki korisnik.

Slika 28. Izgled kada je izabran radnik



Izvor: autor

Pravokutnik sa zelenom pozadinom je prikaz riješenih ticketa od strane izabranog radnika. Plavi pravokutnik prikazuje ukupan zbroj riješenih i odbačenih ticketa, a crveni pravokutnik prikazuje broj odbačenih ticketa.

Slika 29. Izgled modala kada se pritisne na pravokutnik



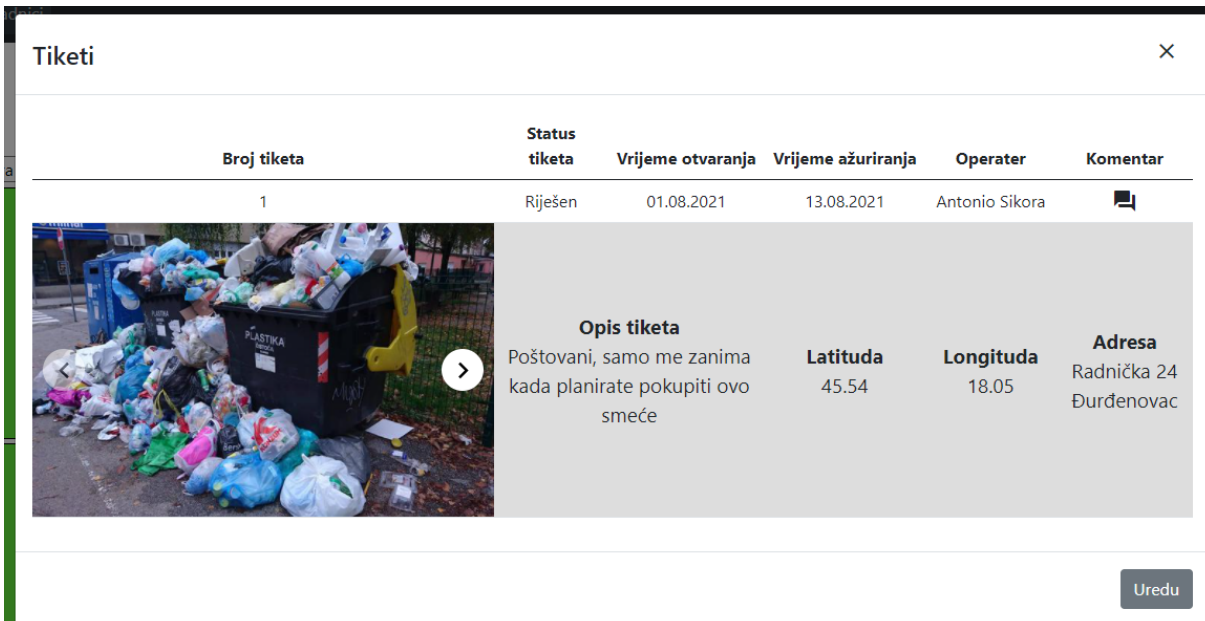
Broj tiketa	Status tiketa	Vrijeme otvaranja	Vrijeme ažuriranja	Operater	Komentar
1	Riješen	01.08.2021	13.08.2021	Antonio Sikora	

Uredu

Izvor: autor

Kada se pritisne na jedan od pravokutnika otvori se tablica s pripadajućim ticketima. Ukoliko pritisnemo na zeleni pravokutnik otvorit će se tablica sa svim riješenim ticketima od strane izabranog radnika. Ako pritisnemo na plavi pravokutnik otvorit će se tablica sa svim ticketima od tog radnika, a ako pritisnemo na crveni pravokutnik otvorit će se tablica s odbačenim ticketima.

Slika 30. Izgled detalja ticketa



Broj tiketa	Status tiketa	Vrijeme otvaranja	Vrijeme ažuriranja	Operater	Komentar
1	Riješen	01.08.2021	13.08.2021	Antonio Sikora	

Opis tiketa
Poštovani, samo me zanima kada planirate pokupiti ovo smeće

Latituda
45.54

Longituda
18.05

Adresa
Radnička 24
Đurđenovac

Uredu

Izvor: autor

Kada pritisnemo na određeni ticket otvore se detalji određenog ticketa. U detaljima imamo slider¹³ sa slikama ukoliko postoje, opis ticketa, latitudu, longitudu i adresu koja je na tim koordinatama. Ukoliko ticket nema unesene slike prikaže se preddefinirana slika.

¹³ slider – galerija slika koju možemo listati

5. ZAKLJUČAK

Prije izrade same aplikacije bilo je potrebno detaljno proučiti zahtjeve aplikacije i napraviti tablice u bazi podataka. Nakon proučavanja zahtjeva i napravljene baze podataka sa svim potrebnim tablicama moglo se početi sa izradom API-ja. API služi za dohvaćanje, spremanje i uređivanje podataka iz prijašnjih napravljenih tablica. Nakon izrade svih potrebnih API-ja bilo je vrijeme za započeti s izradom vizualnog dijela aplikacije u Angular-u. Prije same izrade u Angular-u bilo je potrebno skicirati dizajn same aplikacije.

Znanja na kojima je temeljena ova web aplikacija stečena su na kolegijima studija Računarstva. Također veliki utjecaj imala je i stručna praksa na kojoj je puno toga naučeno o Node.js API-ju i Angular-u.

6. LITERATURA

Internetski izvori:

1. Što je WEB aplikacija <https://geek.hr/pojmovnik/sto-je-web-aplikacija/> (15.08.2021.)
2. Maltar, Jurica; Angular, 2017 <https://zir.nsk.hr/islandora/object/mathos:142/preview> (15.08.2021.)
3. Mujadžević, Edin; Uvod u CSS, 2014
https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/c220_polaznik.pdf (16.08.2021.)
4. Šimić, Ivana; Uvod u TypeScript, 2016 <https://www.mono.hr/2016/12/15/uvod-u-typescript-code-camp/> (16.08.2021.)
5. Tošić, Kosta; Šta je Node.js?, 2015
https://www.popwebdesign.net/popart_blog/2015/06/sta-je-node-js/ (17.08.2021.)
6. W3Schools https://www.w3schools.com/nodejs/nodejs_intro.asp (17.08.2021.)
7. Stančer, Denis; Osnove JavaScripta, 2015
https://www.srce.unizg.hr/files/srce/docs/edu/osnovni-tecajevi/c501_polaznik.pdf (18.08.2021.)
8. https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction (19.08.2021.)
9. Petrovečki, Martina; Kovačević, Željko; Istaknute značajke SQL servera 2019, 2019
<https://hrcak.srce.hr/file/322149> (19.08.2021.)
10. <https://www.tutorialspoint.com/sql/sql-overview.htm> (19.08.2021.)

Nastavni materijali:

1. dr. sc. Oliver Jukić, prof. v. š.: Skriptni programski jezici, Veleučilište u Virovitici
2. mr. sc. Damir Vuk, v. pred.: Baze podataka, Veleučilište u Virovitici

7. POPIS ILUSTRACIJA

Slika 1. Osnovni HTML	3
Slika 2. Primjer CSS koda	4
Slika 3. Primjer definiranja tipa varijable	5
Slika 4. Primjer definiranja tipa funkcije	5
Slika 5. Primjer jQuery koda	6
Slika 6. Komanda za instaliranje pomoću NPM-a	7
Slika 7. Primjer definiranja API rute	8
Slika 8. Primjer dohvaćanja podataka sa SQL-om	9
Slika 9. Primjer spremanja podataka sa SQL-om	9
Slika 10. Primjer brisanja podatka sa SQL-om	10
Slika 11. Arhitektura aplikacije	11
Slika 12. Struktura baze podataka	13
Slika 13. Login	14
Slika 14. Početna stranica	15
Slika 15. Modal statistike	15
Slika 16. Filteri	16
Slika 17. Modal za izbor geografskog područja	17
Slika 18. Modal s komentarom	18
Slika 19. Detalji ticketa	18
Slika 20. Heat mapa	19
Slika 21. Heat mapa kad je uvećano određeno područje	20
Slika 22. Filter za izbor korisnika	20
Slika 23. Filter za starost ticketa	21
Slika 24. Filter po datumu	21
Slika 25. Filter po statusu	22
Slika 26. Navigacija	22
Slika 27. Izbor radnika	23
Slika 28. Izgled kada je izabran radnik	23
Slika 29. Izgled modala kada se pritisne na pravokutnik	24
Slika 30. Izgled detalja ticketa	24



Veleučilište u Virovitici

OBRAZAC 5

IZJAVA O AUTORSTVU

Ja, ANTONIO SIKORA

izjavljujem da sam autor/ica završnog/diplomskog rada pod nazivom

KOMUNALNI ASISTENT - REPORTING WIZARD

Svojim vlastoručnim potpisom jamčim sljedeće:

- da je predani završni/diplomski rad isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu, a što pokazuju korištene bilješke i bibliografija,
- da su radovi i mišljenja drugih autora/ica, koje sam u svom radu koristio/la, jasno navedeni i označeni u tekstu te u popisu literature,
- da sam u radu poštivao/la pravila znanstvenog i akademskog rada.

Potpis studenta/ice

Antonio Sikora



Veleučilište u Virovitici

OBRAZAC 6

**ODOBRENJE ZA POHRANU I OBJAVU
ZAVRŠNOG/DIPLOMSKOG RADA**

Ja ANTONIO SIKORA

dajem odobrenje za objavljivanje mog autorskog završnog/diplomskog rada u javno dostupnom digitalnom repozitoriju Veleučilišta u Virovitici te u javnoj internetskoj bazi završnih radova Nacionalne i sveučilišne knjižnice bez vremenskog ograničenja i novčane nadoknade, a u skladu s odredbama članka 83. stavka 11. Zakona o znanstvenoj djelatnosti i visokom obrazovanju (NN 123/03, 198/03, 105/04, 174/04, 02/07, 46/07, 45/09, 63/11, 94/13, 139/13, 101/14, 60/15, 131/17).

Potvrđujem da je za pohranu dostavljena završna verzija obranjenog i dovršenog završnog/diplomskog rada. Ovom izjavom, kao autor navedenog rada dajem odobrenje i da se moj rad, bez naknade, trajno javno objavi i besplatno učini dostupnim:

- a) široj javnosti
- b) studentima i djelatnicima ustanove
- c) široj javnosti, ali nakon protoka 6 / 12 / 24 mjeseci (zaokružite odgovarajući broj mjeseci).

Potpis studenta/ice

Antonio Sikora

U Virovitici, 30.08.2017

**U slučaju potrebe dodatnog ograničavanja pristupa Vašem završnom/diplomskom radu, podnosi se pisani obrazloženi zahtjev.*