»Mladi za napredek Maribora 2020«

37. Srečanje

MyPark –Mobilna aplikacija za preverjanje parkirnih mest

Inovacijsko področje: aplikativni inovacijski predlogi in projekti

Avtor: Mark Berdnik, Gregor Gril

Mentor: Manja Sovič Potisk

Šola: Srednja elektro-računalniška šola Maribor

Maribor, 2020

»Mladi za napredek Maribora 2020«

37. Srečanje

MyPark –Mobilna aplikacija za preverjanje parkirnih mest

Inovacijsko področje: aplikativni inovacijski predlogi in projekti

Avtor: Mark Berdnik, Gregor Gril

Mentor: Manja Sovič Potisk

Šola: Srednja elektro-računalniška šola Maribor

Maribor, 2020

KAZALO

KAZALO SLIK

1 Povzetek

MyPark je mobilna aplikacija za preverjanje veljavnosti parkirnih mest, kjer uporabnikom prikazuje prosta/zasedena parkirna mesta v njihovi bližini. Ustvarjena je bila s sodobnim ogrodjem Flutter ter programskim jezikom Dart. S pomočjo omrežnega ogrodja DarkNet in računalniške vizije OpenCV, smo ustvarili video analitiko parkirišča, ki uporabnikom v aplikaciji izpiše koliko je prostih parkirnih mest. Za administracijo celotne mobilne aplikacije smo ustvarili spletno stran, ki je namenjena kot nadzorna plošča. Stran smo naredili s tehnologijo MERN (MongoDB, Express.js, React, Node.js) stack. Aplikacija ima tudi dva mikroservisa to sta video analitika in strežnik (Node.js). Dockerizirali smo navedena mikroservisa z uporabo tehnologije Docker.

2 Uvod

Predstavili bomo projekt MyPark, ki smo ga ustvarili z novejšimi tehnologijami. Projekt vsebuje spletno stran, aplikacijo ter računalniški vid. Namen aplikacije je uporabnikom olajšati iskanje prostih parkirnih mest, kar je v današnjem času velik problem, saj ima skoraj vsakdo svoje prevozno sredstvo.

2.1 Cilji:

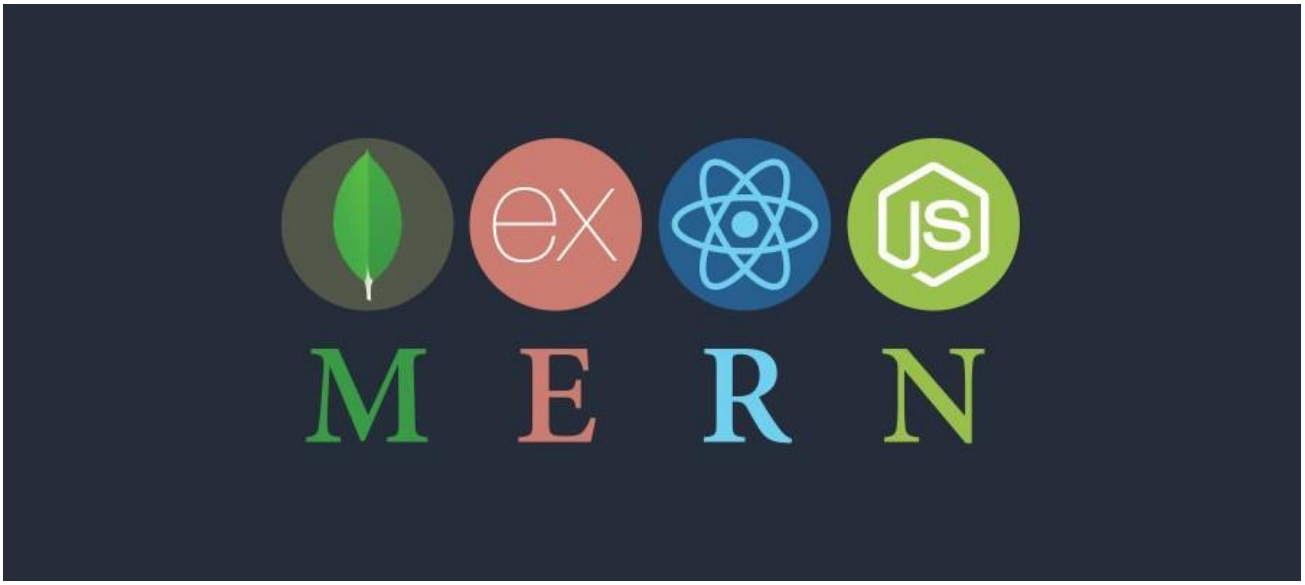
S tem projektom smo si zadali cilj, da bi uporabili novejše tehnologije in se spoznali s njimi, prav tako pa ustvariti uporabno spletno aplikacijo z zasnovano idejo. Aplikacija je povsem dinamična, zato verjamemo, da bi lahko konkurirala z ostalimi mobilnimi aplikacijami v današnjem svetu. Prav tako je predvsem preprosta za uporabo in prijazna starejšim ljudem, ki niso tako vešči o uporabi mobilnih telefonov.

3 Predstavitev

3.1 Programski jeziki

JavaScript:

JavaScript je objektni skriptni jezik, z HTML in CSS je eden izmed treh ključnih tehnologij svetovnega spleta. JavaScript omogoča interaktivne spletne strani in je zato bistven del spletnih aplikacij. Velika večina spletnih strani ga uporablja in vsi večji spletni brskalniki imajo namenski mehanizem JavaScript za njegovo izvajanje. Ima API-je za delo z besedili, nizi, datumi in regularnimi izrazi. Uporabili smo JavaScript pri izdelovanju spletne strani in mobilne aplikacije (MERN).



Slika 1: MERN logotip

Dart:

Dart je splošni programski jezik, ki ga je prvotno razvil Google, kasneje pa ga je Ecma odobrila kot standard. Uporablja se za izdelavo spletnih, strežniških, namiznih in mobilnih aplikacij.

Dart je ustvarjen za hitro univerzalno razvojno kodo, zato je skoraj vse v Flutter-ju napisano z tem programskim jezikom. Omogoča lažje ustvarjanje animacij in prehodov. V Dart-u se najpogosteje uporabljajo Widgeti, Materiali (mobilno UI ogrodje) in Cupertino (IOS UI ogrodje).

Python:

Python je programski jezik, ki ga je ustvaril Guido van Rossum leta 1990. Ima popolnoma dinamične podatkovne tipe, samodejno upravlja s pomnilnikom in podpira funkcionalno, imperativno oziroma proceduralno, strukturirano in objektno orientirano računalniško programsko paradigmo. Zaradi dinamičnih podatkovnih tipov je podoben jezikom Perl, Ruby, Scheme itd... Razvili so ga kot odprtokodni projekt, ki ga je upravljala neprofitna organizacija Python Software Foundation. Python se v glavnem uporablja za računalniško analitiko in razvijanje internetnih aplikacij. V projektu MyPark se je Python uporabil predvsem za video analitiko parkirišč s pomočjo ogrodji DarkNet in OpenCV.

Slika 2: Python logotip

3.2 Predstavitev ostalega orodja

Flutter:

Flutter je odprtokodno ogrodje za razvoj mobilnih aplikacij, ki ga je ustvaril Google leta 2017. Flutter ni jezik, temveč SDK kot Android SDK. Za razvoj med platformami, ki uporabljajo Flutter, je DART uradni programski jezik. Vse v Flutterju je tako imenovani Widget (gradnik), tudi sama aplikacija je Widget.

Zgrajeni so z uporabo sodobnega ogrodja, ki ga navdihuje React. Dart je nov jezik s C-stilom, ki ga je razvil Google. Prvič se je pojavil leta 2007. Uporablja se za razvoj aplikacij Android in IOS kot tudi osnovno metodo za ustvarjanje aplikaciji v Google Fuchsia. Za Flutter smo se odločili zato, saj je novejša tehnologija, ki deluje za dve platformi (Android in IOS) in ponuja hitrejše ter zmogljivejše aplikacije in lepši izgled.

MongoDB

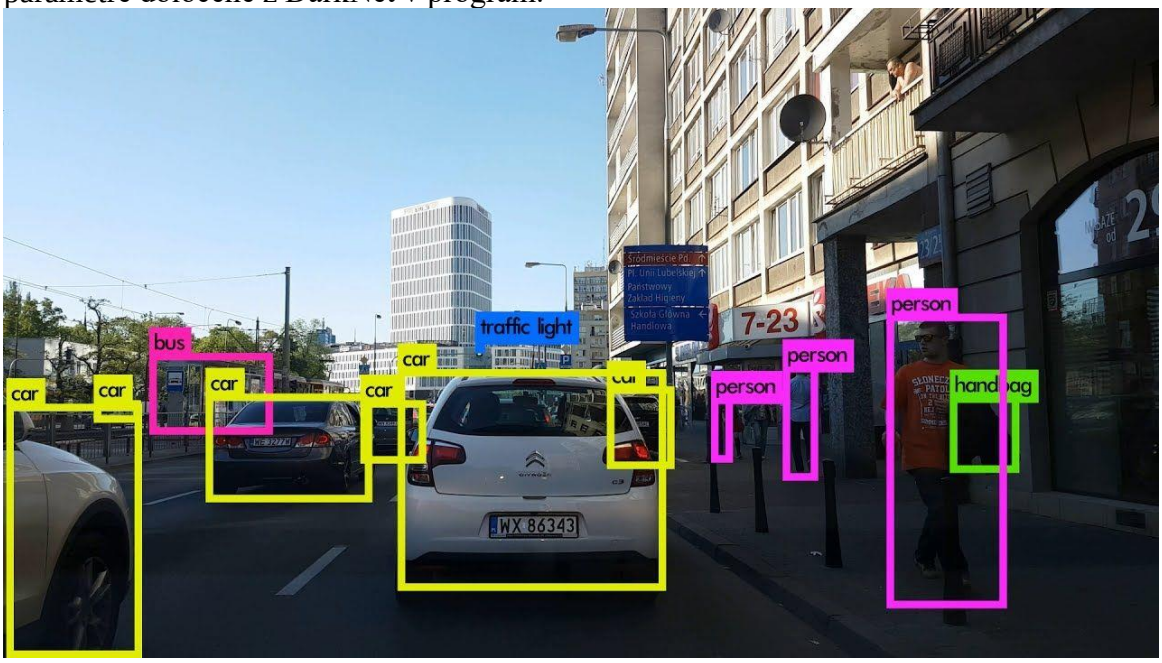
MongoDB je odprtokodni sistem za upravljanje podatkovnih baz, ki uporablja dokumentno usmerjen model baze podatkov. Ta model podpira različne oblike podatkov. Gre za eno od številnih ne-relacijskih tehnologij podatkovnih baz, ki so se pojavile sredi leta 2000 pod zastavico NoSQL.

Za uporabo v velikih podatkovnih aplikacijah in drugih obdelovalnih opravilih, ki vključujejo množice podatke, ki se ne ujemajo s trdim relacijskim modelom. Arhitektura MongoDB je namesto uporabe tabel in vrstic kot v relacijskih bazah podatkov sestavljena iz zbirk in dokumentov.
Izbrali smo jo, saj je hitrejša/zmogljivejša podatkovna baza od večine drugih.

OpenCV

OpenCV je odprtokodna računalniška vizija in knjižnica programske opreme za strojno učenje. OpenCV je bil zgrajen, da bi zagotovil skupno infrastrukturo za aplikacije računalniškega vida in pospešil uporabo strojnega zaznavanja v komercialnih izdelkih. Knjižnica ima več kot 2500 algoritmov, ki vključujejo obsežen nabor klasičnih in najsodobnejših algoritmov za računalniški vid in strojno učenje.

Ti algoritmi se lahko uporabljajo za zaznavanje in prepoznavanje obrazov, identifikacijo objektov, razvrščanje človeških dejanj v videoposnetke, sledenje premikajočim objektom, podobo celotnega prizora, odstrani rdeče oči s slik posnetih z uporabo bliskavice, itd… Knjižnica se pogosto uporablja v podjetjih, raziskovalnih skupinah in vladnih organih. S pomočjo programskega jezika Python, smo z OpenCV implementirali svoje določene parametre določene z DarkNet v program.



Slika 3: Primer uporabe OpenCV

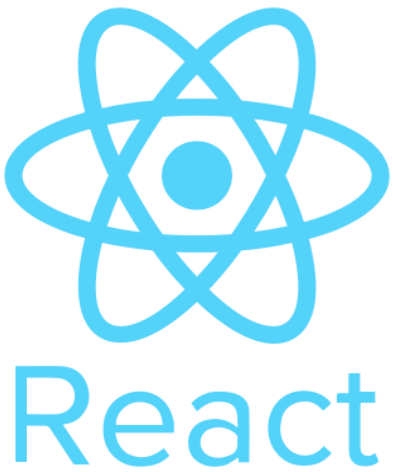
Darknet

DarkNet je odprtokodno živčno omrežno ogrodje, ki se uporablja za strojno učenje računalniškega vida.
Globoko učenje je ključna tehnologija za avtomobile brez voznikov, ki jim omogočajo, da prepoznajo znake ali razločijo pešca od svetilke. V zadnjem času pridobiva veliko pozornosti in to z dobrim razlogom.
Z globokim učenjem se računalniški model nauči izvajati naloge razlikovanja iz slik, besedil ali zvoka.

Modeli učenja lahko dosežejo najboljšo natančnost, ki je včasih boljša kot na človeški ravni. Modeli so usposobljeni z uporabo velikega števila označenih podatkovnih in nevronskih omrežnih arhitektur, ki vsebujejo veliko slojev. Odločili smo se za DarkNet, saj z njim najlažje določimo parametre za parkirna mesta, s katerimi bo računalnik prepoznal ali je mesto prosto/zasedeno.

React

React je JavaScript knjižica za izdelavo uporabniških vmesnikov. Vzdržuje jo Facebook in skupnost posameznih razvijalcev in podjetij. Vse v React-u so komponente, z gradnjo in združevanjem vseh teh komponent se ustvari dinamična spletna aplikacija. Največja prednost uporabe komponent je, da se lahko kadarkoli spremenijo, ne da bi to vplivalo na ostale aplikacije. Ta funkcija je najbolj učinkovita, če se izvaja z večjimi aplikacijami v realnem času, kjer se podatki pogosto spreminjajo.
Vsakič, ko so dodani ali posodobljeni kateri koli podatki, ReactJS samodejno posodobi določeno komponento. To shrani brskalniku nalogo ponovnega nalaganja celotne aplikacije, da odraža spremembe.
Zaradi teh razlogov se nam je React zdel najprimernejša knjižica za izdelavo administracijske spletne strani, saj se bi podatki pogosteje nabirali.



Slika 4: React logotip

NodeJS

Node.Js je odprtokodno JavaScript okolje za izvajanje tega programskega jezika zunaj brskalnika. Node.js je napisal Ryan Dahl leta 2009, približno trinajst let po uvedbi prvega JavaScript okolja na strežniški strani. Prvotna izdaja je podpirala le Linux in Mac OS X. Njegov razvoj in vzdrževanje je vodil Dahl, kasneje pa ga je sponzoriral Joyent.

Razvijalcem omogoča uporabo JavaScripta za pisanje ukazov za skriptiranje na strežniku - izvajanje skriptnih strani strežnika za izdelavo dinamične vsebine spletne strani, preden se stran pošlje v spletni brskalnik uporabnika. Node.js lahko na strežniku ustvarja, odpre, bere, piše, briše in zapira datoteke. Lahko zbira podatke obrazcev in dodaja, briše, spreminja podatke v bazi. Uporabili smo NodeJS za dinamiko spletne strani in z njim ustvarili svoj API.

Slika 5: NodeJS logotip

ExpressJS

ExpressJS je orodje za NodeJS kot brezplačna odprtokodna programska oprema pod licenco MIT. Zasnovan je za izdelavo spletnih aplikacij in API-jev. Ustanovil ga je TJ Holowaychuk, njegova prva izdaja po GitHub, je bila 22. maja 2010. Express zagotavlja vmesnik za izdelavo spletnih aplikacij. Zagotavlja orodja, ki so potrebna za gradnjo aplikacije. Prilagodljiv je, saj so na voljo številni moduli, ki jih je mogoče neposredno priključiti v Express.

Docker

Docker je orodje zasnovano za lažje ustvarjanje, uvajanje in zagon aplikacij z uporabo vsebnikov. Kontejnerji omogočajo razvijalcu, da pakira aplikacijo z vsemi deli, ki jih potrebuje, kot so knjižnice in drugi dejavniki, in jih zbere kot en paket. Za razvijalce to pomeni, da se lahko osredotočijo na pisanje kode, ne da bi se morali ukvarjati s sistemom, na katerem se bo na koncu izvajal. Omogoča jim tudi, da z uporabo enega izmed tisočih programov, ki so že izdelani za izvajanje v Dockerjevem vsebniku kot del njihove aplikacije.

Kontejner je standardna enota programske opreme, ki pakira kodo in vse njene odvisnosti, tako da aplikacija teče hitro in tekoče iz enega računalnika na drugega. Image kontejnerja je lahek in samostojen, izvršljiv paket programske opreme, ki vključuje vse, kar je potrebno za zagon aplikacije: koda, izvajalno okolje, sistemska orodja, sistemske knjižnice in nastavitve.

3.3 Metodologija dela

Projekt je sestavljen iz aplikacije ter spletne strani. Za dinamičnost inovacijskega predloga smo ustvarili podatkovno bazo v MongoDB, kjer se bodo shranjevali potrebni podatki. Za brezhibno delovanje aplikacije smo uporabili NodeJS za izdelavo API-ja, potrebna je bila tudi uporaba vmesnika ExpressJS. Vse skupaj smo povezali s spletno stranjo, ki je bila ustvarjena z JavaScript knjižico React ter tako ustvarili dinamično spletno stran. Aplikacija pa je bila zasnovana z ogrodjem Flutter ter programskim jezikom Dart.

4. Predstavitev spletne strani

4.1 Spletna stran za administracijo

Kot je že bilo navedeno je stran namenjena administraciji celotne aplikacije in predvsem razvijalcem oziroma vzdrževalcem. Celotna spletna stran je narejena z MERN stack (MongoDB, Express, React in NodeJS). Vsebuje nadzorno ploščo, administracijo parkirišč in uporabnikov, celoten zemljevid in urejanje lastnega računa.

4.2 Nadzorna plošča

Nadzorna plošča oziroma *Dashboard* prikazuje potrebne podatke in statistiko o aplikaciji.

Del statistike je dinamičen grafikon ki prikazuje število vseh API klicev, ki jih opravi aplikacija. Druga grafikona prikazujeta število vseh uporabnikov v vseh mesecih in število vseh parkirišč ter kjer se nahajajo (mesto).

Slika 6: Nadzorna plošča

4.3 Parkirišča

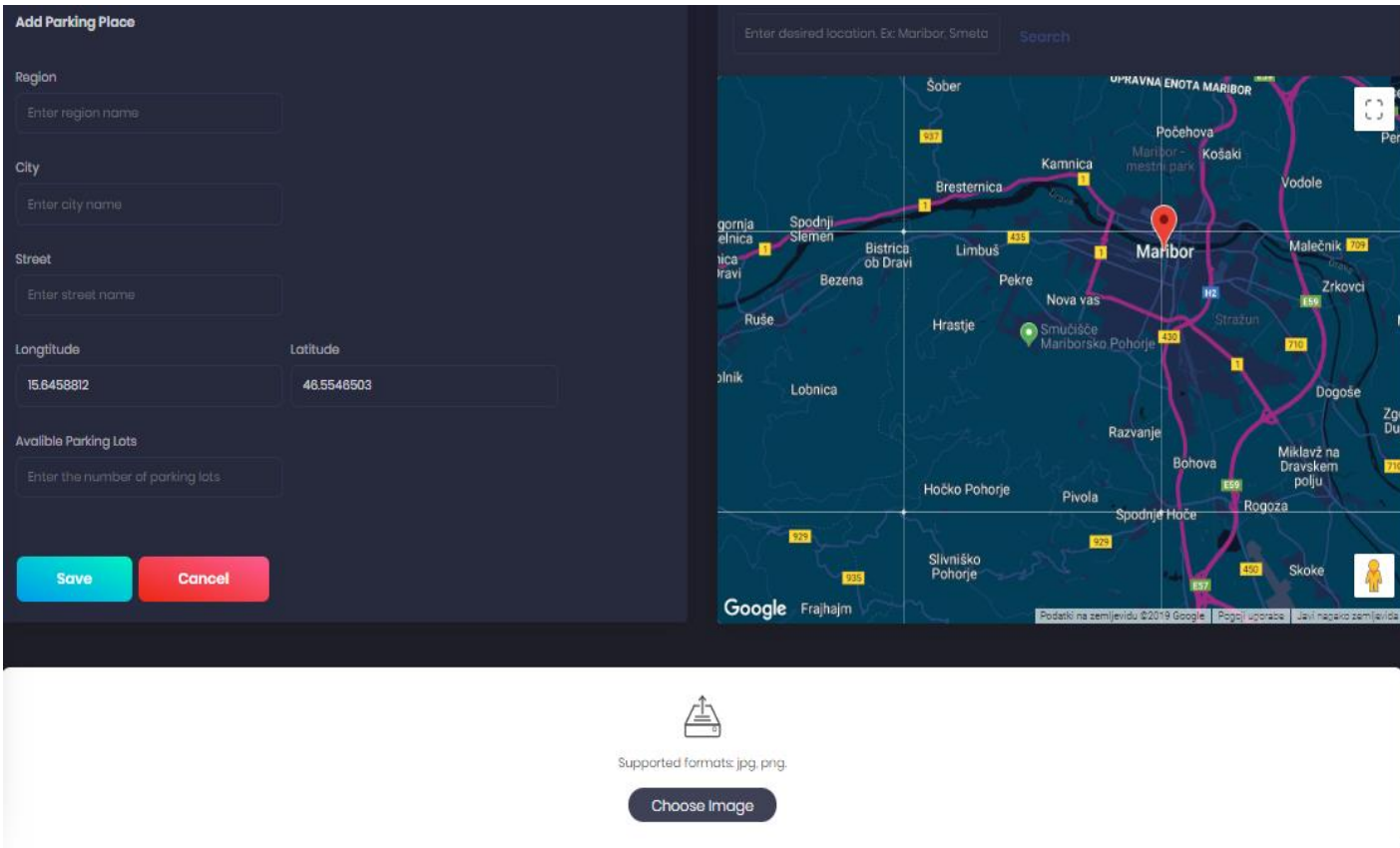
Ker celotna aplikacija temelji okoli parkirnih mest jih je potrebno tudi dodati in imeti nadzor nad podatki.

Prva stran je identična tisti v aplikaciji, torej prikazani so podatki kot so - Regija, Mesto, ulica parkirišča in kljukica/križec ki povesta uporabniku če ima parkirišče prosta oziroma zasedena parkirna mesta.

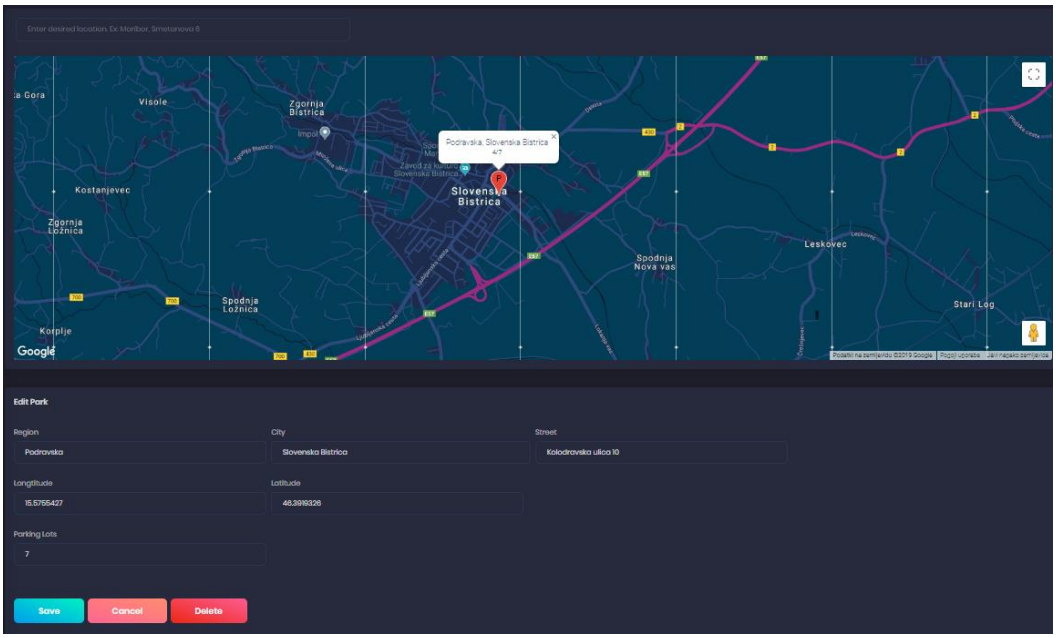
Desno zgoraj smo ustvarili gumb »dodaj parkirišče« (*add park*), stran omogoča administratorju dodajanje vseh potrebnih informacij o določenem parkirišču:

Regijo (Region), mesto (city), ulico (street), longtitude, latitude in število vseh parkirnih mest.

V desnem zgornjem kotu se nahaja iskalna vrstica, če v njo vnesemo pravilno mesto in ulico.



Slika 7: Primer dodajanja parkirišč

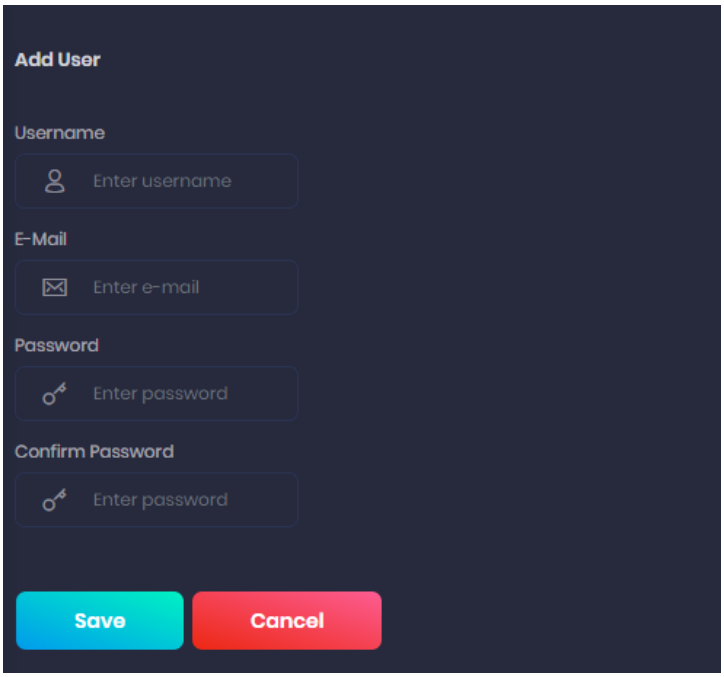


Slika 8: Primer urejanja parkirišč

Za administracijo uporabniških računov smo na strani dodali stran *Users*, ki je namenjena predvsem dodajanju oziroma spreminjanju računov. Na začetni strani so izpisani vsi uporabniki aplikacije, zraven njihovih uporabniških imen, elektronske pošte, skupine in datum ustvarjenega računa je gumb za urejanje.

Na strani se nahaja gumb *add users*, služi za dodajanje uporabniških računov.

Add users je namenjen predvsem za dodajanje administracijskih računov, saj ima aplikacija svoj registracijski sistem.

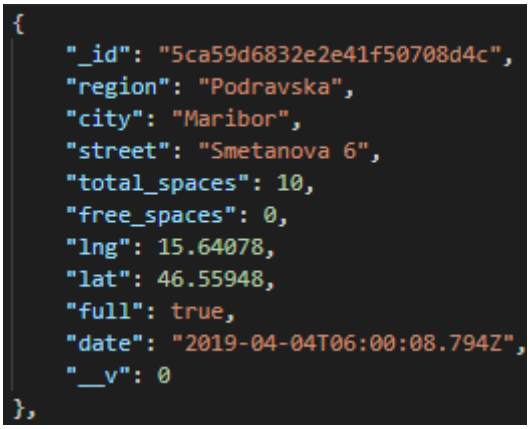


Slika 9: Dodajanje uporabnikov

5. Predstavitev aplikacije

Pričeli smo z izdelavo API-ja, z katerim bi podatki potovali preko strežnika na podatkovno bazo oziroma na aplikacijo. Izdelali smo ga z zgoraj opisanim okoljem NodeJS ki uporablja programski jezik JavaScript. V tem so bile kasneje dodane končne točke (end point), za določene CRUD klice iz aplikacije. Da ima takšen API namen uporabljamo podatkovno bazo MongoDB.

Baza služi shranjevanju vseh potrebnih podatkov za brezhibno dinamično delovanje aplikacije. API shranjuje podatke v obliki JSON formata, v podatkovno bazo. JSON je odprta standardna oblika zapisa datotek, ki za branje podatkovnih objektov, ki jih sestavljajo pari atribut in vrednosti o nizu uporablja človeku berljivo besedilo.



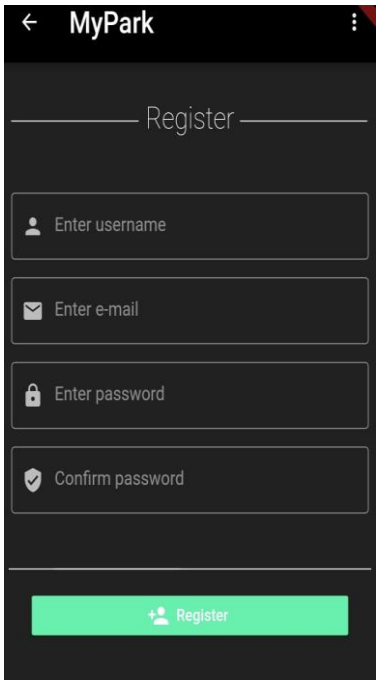
Slika 10: JSON Parks

Zgornja slika prikazuje JSON zapis, v njem je prikazan tako imenovan *Park*. API kliče podatke iz podatkovne baze in jih prikaže na aplikaciji.

Nato smo pričeli izdelovati našo mobilno aplikacijo. Za varnosti in preglednost uporabnikov smo se lotili Login strani (prijava v aplikacijo). Menimo, da je takšen *Login* sistem potreben za pregled statistike (aktivni uporabniki, število vseh uporabnikov) in lažjo administracijo celotne aplikacije.

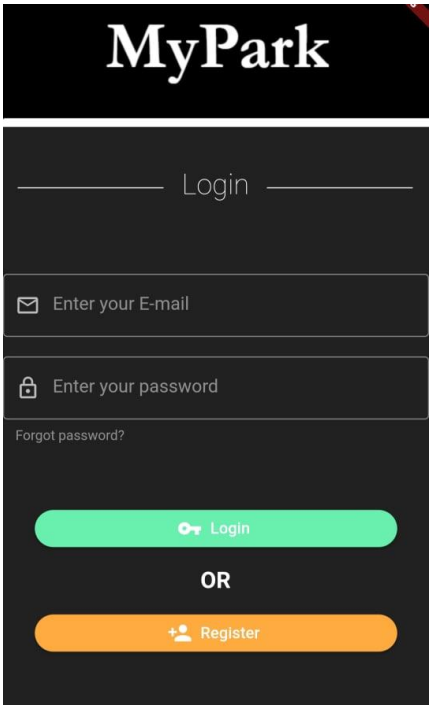
5.1 Login/Register

Da *Login* sistem deluje je potreben *register*, s katerim se podatki shranijo v podatkovno bazo. Oblikovali smo štiri potrebna polja za registracijo – uporabniško ime (*username*), elektronska pošta (*e-mail*), geslo (*password*), potrditev gesla (*confirm password*) in gumb za registracijo (*register*). Ko uporabnik pravilno izpolni vsa polja in pritisne gumb, se podatki preko API-ja shranijo v podatkovno bazo.



Slika 11: Register

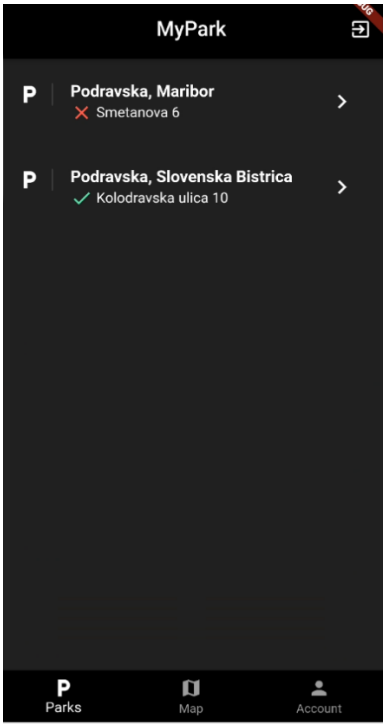
Na prvi strani aplikacije (Login) smo ustvarili dva polja in dva gumba - elektronska pošta (*e-mail),* geslo (*password*) ter gumba login in register. Ob kliku na gumb Login aplikacija preveri če so vpisani podatki že v podatkovni bazi, če so uporabnika vpiše in mu prikaže prvo stran aplikacije.



Slika 12: Login

5.2 Parkirišča

Da pri aplikaciji pridemo do bistva in da uporabnik čimprej najde svoje parkirno mesto, smo se odločili, da bodo na prvi strani aplikacije vsa parkirišča, ki jih podpira aplikacija MyPark. Podatki se na aplikaciji prikažejo preko API-ja iz podatkovne baze, prikazani so samo najpotrebnejši podatki. Za to smo ustvarili dinamično dodajanje parkirišč, na naši administracijski spletni strani jih je mogoče dodajati, urejati in izbrisati. Prikazani so podatki kot so - Regija, Mesto, ulica parkirišča in kljukica/križec, ki povesta uporabniku, če ima parkirišče prosta oziroma zasedena parkirna mesta. S klikom na parkirišče, nas aplikacija preusmeri na informacije o izbranem parkirišču.



Slika 13: Parks

5.3 Dodatne informacije o parkiriščih

Stran dodatne informacije uporabniku izpiše vse podatke o parkirišču ki jih potrebuje.

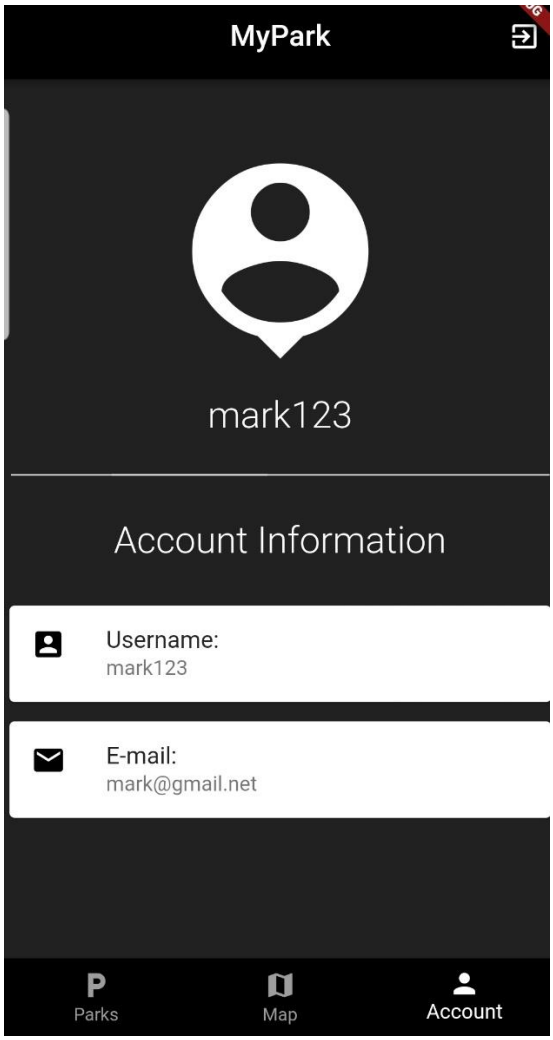
Prikazana je slika parkirišča, ki se vsako minuto posodobi, kamera vsako minuto slika parkirišče, ga predela z video analitiko in prvotno sliko shrani v podatkovno bazo in posledično se slika preko API-ja prikaže na aplikaciji.

Ostale dodatne informacije so – regija (*region*), mesto (*city*), ulica (*street*), število vseh parkirnih mesta (*total spaces*), število prostih parkirnih mest (*available spaces*) in ID parkirišča.

Število prostih parkirnih mest se predela z video analitiko, nato se podatki shranijo v podatkovno bazo in enako kot pri sliki preko API-ja prikaže na aplikaciji.

5.4 Račun

Stran račun smo ustvarili za lažjo preglednost in administracijo za uporabnike. Predvsem je namenjena temu da si uporabniki po želji spremenijo svoje uporabniško (*username*), elektronsko pošto (*e-mail*) in geslo (*password*). Navedene podatke si lahko na strani tudi ogledajo.



Slika 14: Uporabniški račun

5.5 Zemljevid

Zemljevid je namenjen da uporabniku olajšamo oziroma izboljšamo vpogled na lokacijo izbranih parkirnih mest. Uporabili smo Googlov API in tako na aplikaciji prikazali njihovo mapo in na njej so z puščicami označena naša izbrana parkirna mesta.

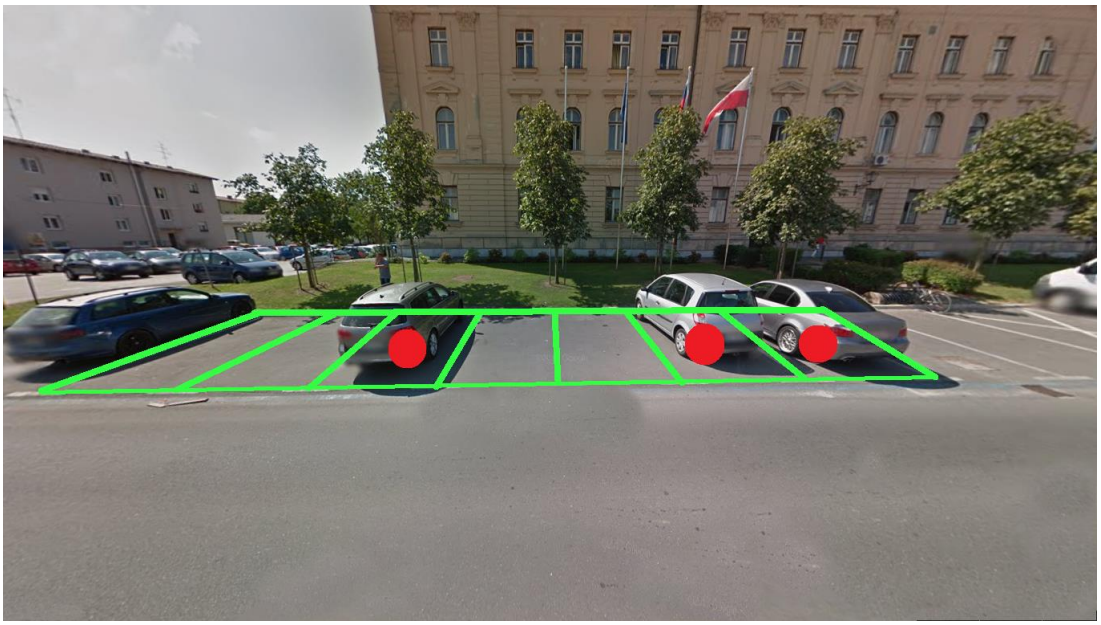
Ob kliku na puščico se prikaže število vseh in število zasedenih parkirnih mest (primer 7/15). S klikom na parkirno mesto aplikacija uporabnika usmeri na stran o dodatnih informacijah.

Slika 15: Zemljevid

6. Python - video analitika

Celotna aplikacija temelji na video analitiki, zato smo morali izdelati svojo delujočo video analitiko parkirišča. Polja smo določili tako, da kamera prepozna, če je parkirno mesto zasedeno oziroma prosto. Rdeča pika nam pove, da je parkirno mesto zasedeno, v nasprotnem primeru polje ostane prazno. Kamera vsako minuto zajame sliko parkirišča, jo z našo analitiko predela in posledično se podatki o predelani sliki preko strežnika prikažejo na aplikaciji.

Slika 16: Primer parkirišča pred video analitiko



Slika 17: Primer parkirišča po video analitiki

7 Teoretični del

7.1 Konkurenca

Konkurenca v današnjem svetu seveda obstaja, vendar v naši državi nismo zasledili aplikacije, ki bi nudila enake funkcije. Obstajajo aplikacije, kot so LPT Parkirišča, ki samo prikažejo obstoječa parkirišča, ne pa prostih parkirnih mest kakor MyPark. V drugih državah obstajajo aplikacije kot so JustPark, ParkME, ParkMobile itd... ki opravljajo podobno delo kot MyPark. Načeloma, kot je komisija omenila, je zelo malo inovacijskih predlogov, ki ponujajo nekaj popolnoma novega, prav tako MyPark ne ponuja nekaj popolnoma neznanega, vendar novost v Sloveniji.

7.2 Prednosti MyPark

Seveda je največja prednost naše aplikacije, kot je že nekaj krat omenjeno, video analitika parkirišč. Torej uporabnikom, bi prihranili veliko živcev ter časa z aplikacijo. Prav tako bi izboljšalo pogled na obstoječa parkirišča, saj bi bila navedena na aplikaciji. Prav tako bi lahko sodelovali z občino, ter olajšali delo tudi redarjem. ki pregledujejo parkirišča. Ena izmed prednosti je tudi preprostost aplikacije za uporabnike majn vešče na mobilnih napravah.

7.3 Slabosti MyPark

Seveda obstajajo v vsaki aplikaciji zraven prednosti tudi slabosti. Aplikaciji MyPark bi lahko imela še več različnih funkcij ter sposobnosti. Ena izmed večjih slabosti je uporaba mobilnih telefonov med vožnjo, katere absolutno ne podpiramo, vendar bi se morali nekako izogniti. Najbolj zaželeno bi bilo, da sovoznik uporabljal aplikacijo ne pa voznik. Menimo tudi, da bi se dalo izgled spletne strani nekoliko izboljšati.

8 Družbena odgovornost

Odgovornost za vpliv

Odgovarjali bi za vpliv aplikacije na okolico in družbo. Menimo, da aplikacija ne bi imela ogromnega negativnega vpliva na okolje, saj ljudem prihrani čas, v drugem pogledu pa je negativna stran majn gibanja.

Preglednost podatkov

Aplikacija ne bi imela negativnega zdravstvenega vpliva na družbo.

Etično obnašanje

Upoštevali bi vse pravice uporabnikov, poskrbeli glede kraj gesel/izgub profilov, in odgovarjali na vprašanja uporabnikov in jim pomagali. V nasprotnem primeru bi bilo potrebno posredovati in ukrepati.

Spoštovanje interesov deležnikov/interesnih skupin

Spoštovali bi interese vseh vključenih delavcev/uporabnikov aplikacije.

Spoštovanje vladavine prava

MyPark bi upošteval vsa pravna načela in zakone republike Slovenije.

9. Zaključek

Video analitika je ena izmed obetavnejših panog računalništva, s pomočjo nje verjamemo, da bo človeštvo v prihodnosti pridobilo še veliko koristi. Globo učenje se že dan danes uporablja v medicini in drugih strokovnih področjih. Menimo da je naša ideja zelo praktičnega pomena in bi lahko bila implementirana v resničnem svetu.

Prva izdaja tehnologije Flutter leta 2017, je bila narejena striktno za izdelovanje mobilnih aplikacij.

V zadnjem času pa napreduje in se izpopolnjuje, da bo lahko v prihodnosti uporabljena tudi za izdelovanje spletnih strani. Omenjena tehnologija uporablja programski jezik Dart, ki ga je ustvaril Google.

Ob pričetku izdelovanja arhitekture dela, nismo bili prepričani, če bomo zmožni celotno aplikacijo uresničiti z vsemi tehnologijami, ki smo si jih izbrali. V poteku izdelave mobilne aplikacije smo se začeli zavedati, da bo projekt bil uspešen. Ustvarili smo celotno aplikacijo z novejšim ogrodjem, uporabili celotno tehnologijo MERN stack in uspešno uvedli eno izmed obetavnejših panog računalništva video analitiko.

10. Literatura

[1] http://lvelho.impa.br/ip08/reading/rt-ocv.pdf, Real-Time Computer Vision with OpenCV, 10.5.2019

[2] https://medium.com/javascript-in-plain-english/full-stack-mongodb-react-node-js-express-js-in-one-simple-app-6cc8ed6de274, Flutter Layout Cheat Sheet, 10.5.2019

[3] https://medium.com/javascript-in-plain-english/full-stack-mongodb-react-node-js-express-js-in-one-simple-app-6cc8ed6de274, Build a full stack MongoDB, React, Node and Express (MERN) app, 10.5.2019

[4] https://blog.paperspace.com/how-to-implement-a-yolo-v3-object-detector-from-scratch-in-pytorch-part-5/, Primer OpenCV slike, 10.5.2019

[5] https://cdn-images-1.medium.com/max/800/1*gqBLqChWtWLq33DvWm6Nog.png, Flutter logotip, 10.5.2019

[6] http://www.jsweet.org/wp-content/uploads/2016/04/react-logo-300x289.png, React logotip, 10.5.2019

[7] https://medium.com/@habibridho/docker-as-deployment-tools-5a6de294a5ff, How to Deploy App Using Docker, 10.5.2019

[8] https://link.springer.com/article/10.1007/s10278-017-9965-6, Toolkits and Libraries for Deep Learning, 10.5.2019