

WebGIS i geovizualizacija lokacija za testiranje na Covid-19 u Republici Hrvatskoj

Tadić, Josip

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:137:500189>

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

Download date / Datum preuzimanja: **2024-12-20**



Repository / Repozitorij:

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli
Fakultet Informatike u Puli

JOSIP TADIĆ

**WEBGIS I GEOVIZUALIZACIJA LOKACIJA ZA TESTIRANJE NA COVID-19 U REPUBLICI
HRVATSKOJ**

Diplomski rad

Pula, rujan, 2021.

Sveučilište Jurja Dobrile u Puli

Fakultet Informatike u Puli

JOSIP TADIĆ

**WEBGIS I GEOVIZUALIZACIJA LOKACIJA ZA TESTIRANJE NA COVID-19 U REPUBLICI
HRVATSKOJ**

Diplomski rad

JMBAG: 0303061400, izvanredni student

Studijski smjer: Informatika

Predmet: Geoinformacijski sustavi

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informacijske i komunikacijske znanosti

Znanstvena grana: Informacijsko i programsko inženjerstvo

Mentor: izv. prof. dr. sc. Ivan Pogarčić

Pula, rujan, 2021.



IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani Josip Tadić, kandidat za magistra Informatike ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mogega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljeni način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

U Puli, 12.09.2021.



IZJAVA O KORIŠTENJU AUTORSKOG DJELA

Ja, Josip Tadić dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom WEBGIS I GEOVIZUALIZACIJA LOKACIJA ZA TESTIRANJE NA COVID-19 U REPUBLICI HRVATSKOJ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, 12.09.2021.

Potpis

Sadržaj

Uvod	1
1. Web	2
1.1. Povijest web-a	2
1.2. Web tehnologije	3
1.2.1. HTML	3
1.2.2. CSS	4
1.2.3. JavaScript.....	6
1.2.4. TypeScript.....	7
1.2.5. React.....	8
1.2.6. JSON.....	9
2. Geografski Informacijski Sustavi.....	12
2.1. Rasterski model	13
2.2 . Vektorski model.....	14
2.3. Koordinatni sustavi.....	16
3. Kartografija.....	18
3.1. Problematika kartografije.....	19
3.2. Mjerila, Simboli, Projekcije	20
3.2.1. Podjela projekcija sa očuvanjem metričkih svojstava	21
3.2.2. Projekcije s obzirom na preslikavajuću površinu.....	25
3.3. Vrste karata	28
3.4. Web kartiranje.....	30
3.4.1. Web Map Service.....	31
3.4.2. Tiled web map	31
3.5. Web karte	32
3.5.1. OpenStreetMap.....	33
3.5.2. MapBox.....	34
3.5.3. Google Maps.....	34
3.6. Razvijanje karti pomoću Web tehnologija.....	35
3.6.1. Leaflet	35
3.6.2. ArcGIS	36
3.6.3. OpenLayers.....	37

3.6.4. Google Maps JavaScript API	38
4. Razvoj aplikacije	39
4.1. Inicijalizacija karte (<i>React-leaflet</i>)	40
4.2. Inicijalizacija alata za crtanje <i>React-leaflet-draw</i>	50
4.3. Geovizualizacija centara za testiranje na CoVid-19.....	52
Zaključak.....	63
Literatura:.....	64

Uvod

Razvoj web tehnologija omogućio je procvat ostalih grana znanosti. Web tehnologije u današnje vrijeme imaju utjecaj na cjelokupni život čovječanstva, pa izuzetak nisu i znanosti. Web tehnologije i Internet svakim danom su u sve više dijelova svijeta, a razni alati i aplikacije imaju sve više korisnika. Ovim napredcima također je potaknut razvoj GIS-a i kartografskih aplikacija. GIS i kartografski alati razvojem web tehnologija postali su praktički sveprisutne, gotovo svaki mobilni uređaj ima određenu aplikaciju koja uključuje nekakav tip karata. Najčešće aplikacije su *Google Maps* i *Apple Maps*. Osim aplikacija koje se koriste za pretrage lokacija, navigacije i slično, GIS i kartografski sustavi primjenjuju se u velikoj mjeri i za analitike raznih vrsta. Razvojem raznih programskih alata područje Geoinformacijskih sustava postalo je dostupno široj javnosti, te je samim time uključen veći broj ljudi u područje. Izrada, prilagođavanje i dijeljenje karata jednostavnije je nego što je to do sada bilo. Postoje brojna rješenja razvijena u svrhu kartiranja i korištenja lokacijskih informacija.

U ovom radu su obrađene teme od web tehnologija, GIS-a i kartografije, do izrade aplikacije koja sjedinjava sva područja. Na samom kraju rada opisan je postupak kako prikazati lokacijske informacije uz pomoć određenih programskih biblioteka. Informacije su u današnje vrijeme vrlo važne za razna područja, stoga je i za očekivati da lokacijske(geografske) informacije od jednake važnosti.

1. Web

1.1. Povijest web-a

1960-tih godina krenulo je razvijanje Interneta u svrhu zadovoljenja ljudske potrebe za dijeljenjem informacija. Tih godina računala su bila glomazna i robusna te je dijeljenje informacija zahtijevalo spremanje podataka na magnetsku vrpcu ili zapisivanje na papir, slanje poštom ili drugu vrstu transporta do drugog računala. Tijekom hladnog rata dodatni razvoj ovakvih tehnologija je bio potaknut te je razvijen ARPANET, iz kojeg je nastala mreža koju danas znamo kao Internet. Datum 1. siječnja 1983. smatra se danom rođenja Interneta jer je ARPANET službeno promijenjen u TCP/IP standard uz pomoć kojeg je preko mreže bila omogućena komunikacija između dva računala (University System of Georgia).

Ovo poglavlje izuzetno je bitno kako bi se razjasnila razlika između Interneta i weba. Internet je globalna umrežena veza mreža, dok je *world wide web* kolekcija informacija kojom se pristupa preko Interneta. Skraćeno, Internet je infrastruktura, a Web je servis sagrađen na toj infrastrukturi(GeeksforGeeks).

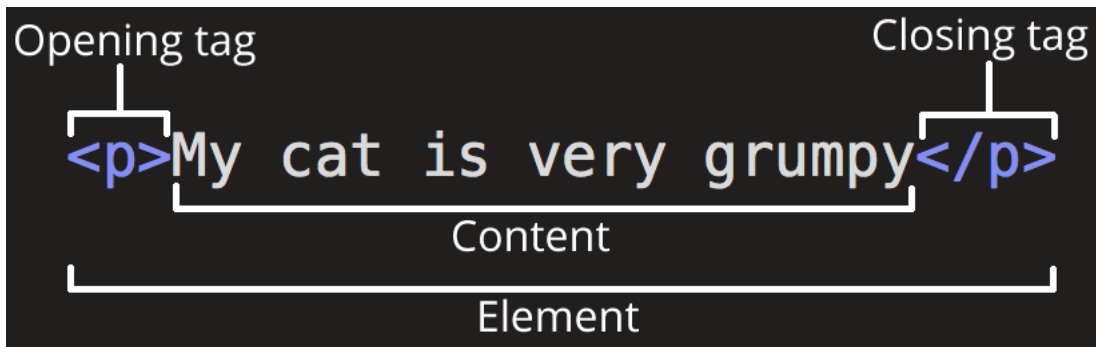
World Wide Web ili skraćeno WWW izmislio je britanski znanstvenik Tim Berners-Lee 1989 godine. Za to vrijeme radio je u CERN-u, a potreba za *world wide webom* javila se kako bi se omogućilo automatizirano dijeljenje informacija između znanstvenika širom svijeta. Osnovna ideja weba bila je sjedinjenje brzo razvijajućih računalnih tehnologija, podatkovnih mreža i takozvanog „hiperteksta“ u jedan globalni sustav jednostavan za korištenje. U prosincu 1991. godine prvi Web server u Sjedinjenim Američkim Državama je postao dostupan. Kada je 30. travnja 1993. godine CERN javno objavio programski kod Weba, te kada je Web postao besplatan softver, samo do kraja te godine već je bilo preko 500 poznatih web servera. Taj broj je 1994. godine narastao na 10000 servera i preko 10 milijuna korisnika. 1994. godine Berners-Lee napustio je CERN kako bi se pridružio MIT-u da bi osnovao „International World Wide Web Consortium“ (W3C) (CERN).

1.2. Web tehnologije

Web tehnologije u današnje vrijeme pokreću svijet, a razlog iz kojega se javila potreba u ovom radu obraditi tu temu leži upravo u njenoj rasprostranjenosti. Web stranice ili web aplikacije najčešće se pokreću u programima specijaliziranim za pregled i rad sa njima, a one se nazivaju preglednicima (eng. *Browser*). Još jedan razlog zašto su web tehnologije odabrane kao pravo rješenje je u tome što danas gotovo svaki uređaj u sebi sadrži neku vrstu preglednika, što znači da je dovoljno razviti jedan sustav ili aplikaciju koja se onda može pokretati na svim uređajima na primjer: stolna računala, laptopi, mobiteli, tableti, i sl. koji koriste web preglednik. Iako preglednici razvijeni od strane drugih proizvođača se ponešto razlikuju, moguće im je prilagoditi aplikacije uz sitne promjene napravljene u razvijanju. Ono što je još jedna prednost je da određeni dijelovi ili čak cijele aplikacije se mogu koristiti u preglednicima bez internetske veze. Web tehnologije osnovane su na određenim standardima, a neki od njih objašnjeni su u narednim poglavljima. Kako bi web stranice funkcionirale ispravno potrebno je držati se navedenih standarda, a standardi su opisani u njihovim službenim dokumentacijama (GeeksforGeeks).

1.2.1. HTML

HTML (eng. *Hyper Text Markup Language*) je jezik za označavanje u kojem se izrađuju web stranice. Web preglednici prikazuju stranice uz pomoć HTML-a. HTML dokumenti se mogu nalaziti na serveru (oblaku) ili lokalno, a način funkcioniranja je isti, izuzev toga što je za „Cloud“ rješenja potrebna Internet veza. Web stranice sastoje se od HTML elemenata koje preglednik formira kao određenu web stranicu. HTML element sastoji se od oznaka (eng. *Tags*) koje se otvaraju i zatvaraju sa znakovima „<“ i „>“. Postoje elementi koji moraju imati svoj zatvoreni par i oni koji završavaju u jednoj oznaci (Mozilla).



Slika 1: primjer HTML tag-a

Izvor: https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started

Primjer jednostavne HTML stranice može izgledati poput:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Primjer jednostavne web stranice</title>
  </head>
  <body>
    <h1>Hello world!</h1>
  </body>
</html>
```

Slika 2: primjer jednostavne HTML stranice

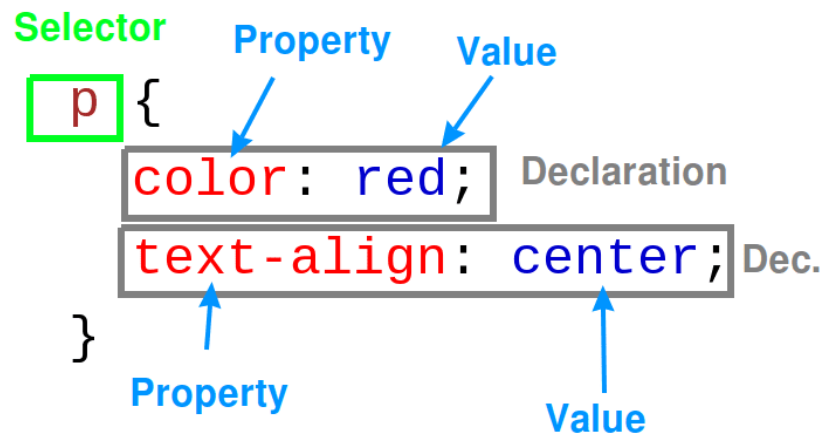
1.2.2. CSS

CSS(eng. *Cascading Style Sheets*) je standard definiran od strane W3C koji se koristi za stiliziranje HTML dokumenata, ali i drugih XML(eng. *Extensible Markup Language*) dokumenata. CSS također služi za različite načine prikazivanja web stranica kako bi bile

što preglednije. Pomoću CSS-a također se prilagođava prikaz u ovisnosti o uređaju (na primjer: stolna računala, laptopi, mobiteli, tableti, i sl.) na kojem se web stranica pregledava. Iako postoji više načina dodjeljivanja CSS vrijednosti, u osnovi se svodi na dodjeljivanje vrijednosti na određena svojstva(eng. *Properties and values*). Unutar selektora (eng. *Selector*) dodjeljuju se željene vrijednosti(Niederst Robbins, 2018, p 239)

Postoji više vrsta selektora, a osnovni su:

1. „*type/element selector*“: odnosi se na određeni HTML element.
2. „*#id selector*“: odnosi se na HTML element naznačen sa odgovarajućim „id“ svojstvom.
3. „*.class selector*“: odnosi se na sve HTML elemente naznačene sa odgovarajućim „class“ svojstvima(Mozilla).



Slika 3: struktura CSS pravila

Izvor: <https://io.bikegremlin.com/12802/html-css-introduction/>

```
h1 {  
  color: red;  
  font-size: 48px  
}
```

Slika 4: primjer CSS pravila za stiliziranje za tag <h1>

Kako bi se CSS pravila primijenila na HTML dokument potrebno ih je povezati, a to se može učiniti na više načina:

1. „*External style sheet*“ je vanjska datoteka sa ekstenzijom .css, koja se mora povezati ili uvesti u HTML dokument. Moguće je i povezivanje jedne CSS datoteke na više HTML dokumenata.
2. „*Embedded style sheet*“ dodaje se na način da u <head> tagu napravimo HTML tag naziva „*style*“ unutar kojega se onda pišu CSS pravila.
3. „*Inline style*“ je način dodjeljivanja CSS pravila unutar pojedinoga HTML taga, a radi se na način da se u HTML tag doda svojstvo „*style=*“, unutar kojega se onda dodjeljuju pravila (Niederst Robbins, 2018, p 245).

1.2.3. JavaScript

JavaScript je skriptni programski jezik uz pomoću kojega se mogu raditi interakcije i kompleksniji elementi web stranice, kao i definirati određena ponašanja HTML i CSS elemenata. European Computer Manufacturers Association (ECMA) je standardizirala JavaScript, stoga se u raznim literaturama i dokumentacijama još uvijek može pronaći naziv „*ECMAScript*“. JavaScript ne postoji samo u kontekstu web-a, iako je u webu veoma zastupljen. JavaScript danas ne služi samo za klijentsko okruženje, već postoji niz alata i tehnologija koje omogućuju razvoj „*Full-stack*“ web aplikacija. Na taj način razvoj web aplikacija postao je jednostavniji jer se jednim programskim jezikom obuhvaća više cjelina (Mozilla).

JavaScript je poznat kao treći sloj web aplikacije, a prva dva sloja su već spomenuta: HTML kao strukturni i CSS kao prezentacijski sloj. Kako za CSS tako i za *JavaScript*, postoji više načina kako pisati skripte pomoću kojih se unaprjeđuju web stranice: (Niederst Robbins, 2018, p 595)

1. „*Embedded script*“: dodaje se na način da se programski kod piše unutar <script> tagova

2. „*External script*“: također se dodaje uz pomoć `<script>` tagova, ali je potrebno definirati svojstvo „src“ u kojemu navodimo putanju do vanjske skripte u kojoj se nalazi programski kod (Niederst Robbins, 2018, p 597)

1.2.4. TypeScript

„*TypeScript* je jezik otvorenog koda (eng. *Open-source*) koji se nadovezuje na *JavaScript*, jedan od najkorištenijih alata na svijetu, dodavanjem statičkih definicija tipova.“ (Microsoft). Razvijen je od strane *Microsoft*-a. Potreba za *TypeScript*-om se javila kako bi olakšala rad sa *JavaScript*om, na način da isprave njegove slabe točke. Najvažnije značajke *TypeScripta* jesu da se tipovi u programskom kodu provjeravaju prije samog izvršavanja programa, zbog čega se *TypeScript*-u pripisuje da je „*Static typechecker*“ (Microsoft).

Značenje ovog pojma demonstrirano je na primjeru:

JavaScript objekt „user“, sadrži dva svojstva: „name“ i „age“, sa vrijednostima: „Daniel“ i „26“, ako pokušamo pozvati metodu „location“ nad objektom user, *JavaScript* vraća „undefined“.

```
const user = {
  name: "Daniel",
  age: 26,
};

user.location; // returns undefined
```

Slika 5: poziv metode „location“ nad objektom „user“ u JavaScriptu

Izvor: <https://www.typescriptlang.org/docs/handbook/2/basic-types.html>

Ukoliko je taj poziv metode nad istim objektom pozvan u *TypeScript*-u, program bi rezultirao greškom jer metoda „*location*“ nije definirana(Microsoft).

```
const user = {  
  name: "Daniel",  
  age: 26,  
};  
  
user.location;
```

Property 'location' does not exist on type '{
name: string; age: number; }'.

Slika 6: poziv metode „*location*“ nad objektom „*user*“ u *JavaScriptu*

Izvor: <https://www.typescriptlang.org/docs/handbook/2/basic-types.html>

1.2.5. React

React je deklarativna i fleksibilna JavaScript biblioteka(eng. *Library*) za izgradnju korisničkih sučelja. Omogućuje sastavljanje složenih korisničkih sučelja od malih i izoliranih dijelova programskog koda koji se nazivaju komponente. *React* koristi tip datoteke koji se naziva „*JSX*“, a izgleda kao mješavina *HTML*,*XML* i *JavaScript/TypeScript* elemenata. *JSX* omogućuje uglavljivanje *JavaScript/TypeScript* programskog koda između ovih elemenata unutar znakova „{, i „}“. Na ovaj način olakšano je rukovanje varijablama i njihovim vrijednostima, ali je i poboljšana preglednost programskog koda. Glavne značajke *React* ostvaruje kroz takozvani „*State*“ i „*Props*“. *State* označava stanje koje je sadržano u određenoj komponenti, a *Props*-i označavaju podatke(vrijednosti) prosljeđene u druge komponente. Nakon svake izmjene u *State*-u *React* odmah prikazuje iste na način da ponovno prikaže komponentu sa novim vrijednostima (Facebook).

Ovakav način rada vrlo je efikasan za izradu korisničkih sučelja, ali i kompleksnijih web aplikacija. S obzirom da se *TypeScript* nadovezuje na *JavaScript*, omogućeno je i korištenje *TypeScript*-a unutar *React* sustava.

1.2.6. JSON

JavaScript Object Notation ili skraćeno JSON je standard za izmjenjivanje podataka, prepoznat od strane IETF(Internet Engineering Task Force) i Ecma International-a. JSON je usvojen zbog jednostavnosti i lakoće čitanja. JSON kao takav je potpuno neovisan o programskom jeziku, a zbog svog tekstualnog formata i strukture koristi se sa većinom poznatijih programskih jezika: JavaScript, Python, C, C#, C++, Java i mnogih drugih. Srž JSON-a jesu dvije strukture: skup imena i vrijednosti, i liste vrijednosti. Prva struktura u programskim jezicima najčešće se ostvaruje kroz objekte, rječnike, hash tablice i liste s ključevima, dok se druga struktura najčešće ostvaruje kroz: liste, vektore i polja. Objekt je skup imena i vrijednosti, te počinje sa znakom „{,“ i završava sa znakom „}“ . Ime i vrijednost su odvojeni znakom „:“, a njihovi parovi se odvajaju znakom „ , “ . Početak polja označava se znakom „[,“ a završetak znakom „]“, a njegove vrijednosti također su odvojene znakom „ , “ . Ovakve strukture široko su poznate i usvojene u raznim programskim jezicima, te iz tog razloga skoro svi programski jezici mogu koristiti JSON format za razmjenu podataka(ECMA).

```
{  
  "name": "John",  
  "age": 30,  
  "cars": ["Ford", "BMW", "Fiat"]  
}
```

Slika 7: jednostavni JSON objekt, sa poljem „cars“.

Izvor: https://www.w3schools.com/js/js_json_arrays.asp

1.2.6.1. GeoJSON

Internet Engineering Task Force definira GeoJSON kao format razmjene geoprostornih podataka temeljen na JSON-u. GeoJSON definira nekoliko vrsta JSON objekata i način na koji se kombiniraju za predstavljanje podataka o zemljopisnim obilježjima, njihovim svojstvima i prostorni opseg. GeoJSON koristi geografski koordinatni referentni sustav, World Geodetic System 1984. i jedinice decimalnih stupnjeva. Koristi se za raznovrsne strukture geografskih podataka. Objekt GeoJSON-a se može predstaviti kao:

1. Područje prostora (eng. „*region of space*“) – „Geometry“.
2. Prostorno ograničen entitet (eng. *spatially bounded entity*) – „Feature“.
3. Lista značajki (eng. „*list of features*“) – „FeatureCollection“.

Geometrijski tipovi podržani u GeoJSON formatu su sljedeći: „Point“, „LineString“, „Polygon“, „MultiPoint“, „MultiLineString“, „MultiPolygon“ i „GeometryCollection“ (Internet Engineering Task Force).

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {},
      "geometry": {
        "type": "Point",
        "coordinates": [
          15.98012924194336,
          45.81217036039273
        ]
      }
    }
  ]
}
```

Slika 8: primjer GeoJSON objekta za jednu točku

Izvor: <http://geojson.io/>

Pozicije (koordinate) nalaze se u polju sa imenom „*coordinates*“, a ovisno o tipu može biti jedna vrijednost ili više. Element na prvom indeksu u polju predstavlja geografsku širinu, a element na drugom indeksu predstavlja geografsku dužinu, postoji i treći opcionalni indeks koji predstavlja visinu (Internet Engineering Task Force).

2. Geografski Informacijski Sustavi

Geografski Informacijski Sustav (eng. *Geographic information system*) ili skraćeno GIS je pojam koji datira iz šezdesetih godina prošlog stoljeća. Iako postoje razne definicije geografskih informacijskih sustava, one su uglavnom ovisne o kontekstu. Prije definiranja GIS-a, pojašnjen je pojam geografskih informacija. Geografske informacije u najširem smislu se odnose na bilo koje informacije vezane za lokacije ili mjesta. U stvarnosti ova definicija je toliko opsežna da bi se geografske informacije moglo vezati uz gotovo sve informacije. Iz tog razloga u obzir se uzimaju samo informacije relevantne uz proučavani problem. Primjeri geografskih informacija mogu biti raznih vrsta: kao prvi primjer razmatrane su granice općina i gradova kao geografske informacije administrativnog tipa, ili pak pozicije cesta i raskrižja kao geografske informacije prostorno prometnog tipa. Postoji mnoštvo drugih primjera koji su vezani uz geografske informacije: popis stanovništva, stopa rodnosti i smrtnosti, broj turista itd. Također, vezano za situaciju s korona virusom: broj zaraženih po gradovima, županijama i državama. Geografske informacije također mogu biti kvalitativnog tipa kao što su naprimjer crteži i fotografije. Ako točna lokacija u informaciji nije dostupna, to ne znači da se ona ne može povezati sa geografskom informacijom, no pitanje je za što će se takva informacija koristiti (Gregory et Eil, 2007, p 3).

Obzirom da je u prethodnom poglavlju obrađena tema web tehnologija, GIS je definiran kao vrsta softvera. Pojednostavljeno, GIS-a definiran kao softver jest alat uz pomoću kojega prikazujemo geografske informacije. Ovakav softverski GIS sustav omogućuje prikaz i rukovanje raznim informacijama na površini zemlje, no valja napomenuti da GIS generalno nije vezan strogo uz zemlju.

Najčešće ovakve informacije su spremljene u bazi podataka, iako to nije uvijek primjer. Rukovanje informacijama iz baze podataka realizira se kroz „Sustav za upravljanje bazom podataka“ (eng. *Database management system (DBMS)*) (Gregory et Eil, 2007, p 3).

Podatci u GIS sustavima se dijele na prostorne i atributne. Prostorni podaci jesu lokacije (na primjer koordinate u koordinatnom sustavu), a atributni podaci se vežu na

njih, i sadrže opisne informacije. Prostorni podaci mogu biti točke, linije ili površine, a atributni podaci najčešće se prikazuju kao „Label“, ali im to nije primarna funkcija(PMR).

Ovi podaci mogu se implementirati uz pomoć relacijske baze podataka. Prostorne značajke u GIS-u mogu se prikazati sa jednim od dva modela: Rasterski i Vektorski(Harvey, 2008, p 176)

2.1. Rasterski model

Rasterski model podsjeća na digitalnu fotografiju. Prikaz se radi na način da se područje karata ili prikaza podjeli na mrežu (najčešće jednakih) piksela ili ćelija, u kojoj se pohranjuje vrijednost(RGB, Grayscale ili sl.) . Što je veći prikaz to se detalji slabije mogu istaknut, dok je kod manjeg prikaza moguće prikazati puno više detalja(Buckey, GIS Introduction).

Prednosti rasterskog modela:

1. Jednostavna podatkovna struktura
2. Preklapanje i kombinacija karata se jednostavno izvodi
3. Nekim od metoda analize pogoduje rasterski prikaz
4. Jednostavne simulacije
5. Jeftina tehnologija
6. Olakšano mjerenje iz zraka(sateliti, letjelice)
7. Moguće je jednostavno prikazati visinsku komponentu
8. Pohrana kao oblik fotografije

Nedostatci rasterskog modela:

1. Nedovoljan prikaz detalja većeg područja
2. Grube raster karte su manje „ugodne oku“

3. Kompliciraniji rasteri zahtijevaju više memorije
4. Prijelaz između područja izgleda „grubo“
5. Teži prikaz topoloških odnosa
6. Kompresijom većih područja dio informacija se može izgubiti(Buckey, GIS Introduction)

2.2. Vektorski model

Vektorski model definiran je matematičkim funkcijama, a njegovi podaci koriste točke sa njihovim x, y koordinatama za konstrukciju prostornih prikaza. Vektorski model podataka kao osnovu koristi vrste objekata: točke, linije i površine. Razlika u odnosu na rasterski model je u tome da iscrtavanja se jednako dobro prikazuju neovisno o rezoluciji. U vektorskom modelu podataka svaka prikazana točka ima jedinstvenu lokaciju određenu koordinatama(Buckey, GIS Introduction).

Prednosti vektorskog modela:

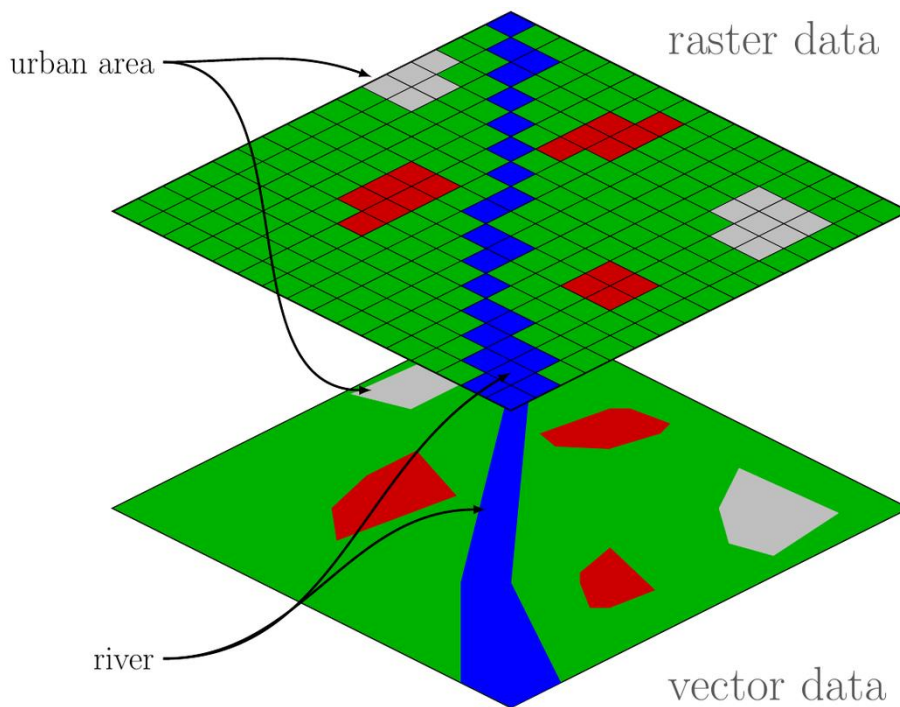
1. Vektorski prikaz je ugodniji ljudskom oku, jer nema „grubih“ dijelova
2. Topologija je jednostavna za održavanje
3. Vrlo precizni prikazi grafike
4. Jednostavnije održavanje, dohvat i ažuriranje atributa
5. Manja memorijska zahtjevnost
6. Lakša kompresija homogenih područja
7. Bolji krajnji rezultat izrade karata

Nedostatci vektorskog modela:

1. Podatkovna struktura je kompleksnija od rasterskog modela
2. Prekomjerno preklapanje višestrukih prikaza može stvoriti poteškoće
3. Simulacije se odrađuju teže zbog raznih topoloških oblika

4. Krajnje mape i ispis istih može biti skuplje zbog razine detalja, pogotovo za visoku kvalitetu
5. Nema kompatibilnosti sa udaljenim prikazima kao kod rasterskog modela
6. Ukoliko postoji visoka varijabilnost, teže je efikasno skladištenje podataka
7. Nemogućnost pohrane fotografskih podataka
8. Analiza zahtjeva više vremena od rasterskog modela(Buckey, GIS Introduction).

Usporedba vektorskog i rasterskog modela:



Slika 9: primjer rasterskog i vektorskog modela prikaza

Izvor: <http://www.eo4geo.eu/training/urban-heat-islands-basic-gis-knowledge-vector-and-raster-data/>

2.3. Koordinatni sustavi

Kako bi geografske informacije pomoću koordinata bile prikazane na karti, potreban je koordinatni sustav, neovisno o mediju. Potreba za koordinatnim sustavom javila se iz razloga što geoprostorne informacije se moraju na neki način referencirati se u prostoru. Ovaj proces se još naziva georefenciranje ili geokodiranje. S druge strane koordinatni sustav je sustav u kojemu uz pomoć brojnih vrijednosti, koje još nazivamo i koordinate, se mogu jednoznačno odrediti poziciju neke točke (Albrecht, 2007, p 2).

Postoje razni koordinatni sustavi a jedan od najjednostavnijih poznat je kao brojevni pravac. Na brojevnom pravcu postoje jedinstvene točke koje su pridružene nekom realnom broju, a osnova brojevnog pravca je pravac. Brojevni pravac kao i svaki drugi koordinatni sustav mora imati ishodište koje je predstavljeno brojem 0. Nakon određivanja ishodišta, na brojevnom pravcu dužina udaljenosti točaka se određuje jediničnom duljinom, a ona predstavlja razmak između dvije točke. To znači da između određenog broja točaka postoji točno određena dužina. Na brojevnom pravcu vrijednosti s desne strane ishodišta su pozitivni, a s lijeve negativni. Sljedeći primjer koordinatnog sustava naziva se Kartezijev koordinatni sustav ili pravokutni koordinatni sustav, a naziv je stekao po francuskom matematičaru Reneu Descartesu koji je i razvio ovaj sustav. U Kartezijevom koordinatnom sustavu u ravnini kroz ishodište prolaze x i y osi koje se sijeku pod pravim kutom. Za jednoznačno određivanje pozicije koristi se par realnih brojeva, sa x i y koordinatom. Jedinstvena dužina određena je na isti način kao i brojevnog pravca, zasebno za svaku os koordinatnog sustava. Na isti princip u Kartezijevom koordinatnom sustavu mogu se odrediti i koordinate točaka u trodimenzionalnom prostoru, ali kroz ishodište tada prolaze tri osi pod pravim kutom, a koordinate točaka su u ovom slučaju (x, y, z). Izum i razvoj Kartezijevih koordinata izazvao je revoluciju u matematici, kreirajući vezu između Euklidske geometrije i algebre. Za primjer se može uzeti krug sa promjerom veličine 3 sa središtem u ishodištu. Takav krug u Kartezijevom koordinatnom sustavu sa svim njegovim točkama u prostoru je moguće prikazati funkcijom $X^2 + Y^2 = 9$.

Kosokutni koordinatni sustav sličan je pravokutnome koordinatnom sustavu ali za razliku od pravokutnog, x i y osi se ne sijeku okomito.

U Polarnom koordinatnom sustavu pozicije točke određuju se pomoću kuta između referentnog pravca(polarnog pravca) i udaljenosti od ishodišta.

Geografski koordinatni sustav je koordinatni sustav uz kojeg se vežu pozicije na Zemlji(Hargitai, 2019, p 80).

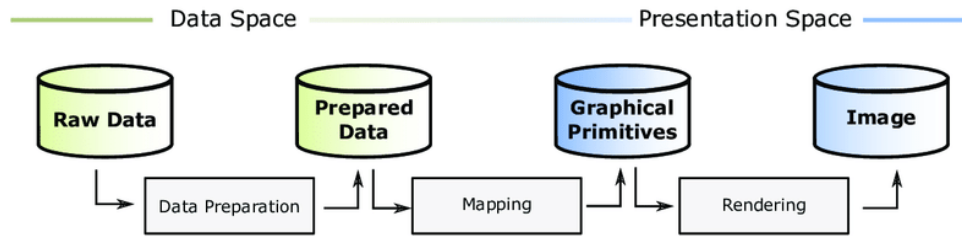
3. Kartografija

Kartografija je tehnika, znanost i umjetnost filtriranja i kompiliranja prostornih podataka u informacije na karti. Postoje razne definicije kartografije, a bitno je naglasiti da kartografija predstavlja cijelo okruženje i sve neophodne alate čiji će krajnji rezultat biti karta. Karte su vrlo važan alat za razumijevanje i analizu geoprostornih podataka. Geoprostorni ili geospacijalni podaci, kao i kartografski alati, dodaju vrijednosti na temu razvijanja kartografije. Bitno je naglasiti da se kartografija kao znanost dotiče mnogih područja, pa je tako svoj utjecaj ostavila i u razvijanju softvera. Iako se uvijek veže uz planet Zemlju na kojoj postoji određena problematika prilikom prikaza geoprostornih podataka na karte, kartografija se također može vezivati i na šire područje kao što su ostali planeti, zvijezde i svemir. Prve naznake ovoga tipa kartografije, koji se naziva i planetarna kartografija, počela se razvijati tijekom razvoja prvih teleskopa, no i prije toga prilikom promatranja zvijezda (Hargitai, 2019, p 3).

Prikaz bilo kakvih podataka, generalno, olakšava njihovo razumijevanje na način da se putem prikaza odnos podataka (ili njihovo kompleksno ponašanje) pojednostavi. Isto je i sa prostornim podacima, a prema (Haber, McNabb(1990), Carpendaleu(2003)) taj proces se može podijeliti na 4 djela:

1. Priprema podataka i transformacija
2. Vizualno mapiranje
3. Generiranje pregleda
4. Percepcija/spoznaja od strane korisnika

Navedeni proces naziva se još „cjevovod vizualizacije“ (eng. „*visualization pipeline*“).



Slika 10: proces prikaza prostornih podataka

Izvor: https://www.researchgate.net/figure/The-Visualization-Pipeline-adapted-from-Haber-and-McNabb-1990_fig7_342311651

3.1. Problematika kartografije

Kartografija kao znanost oduvijek je rješavala određene prostorne probleme, kao na primjer:

1. Predstavljanje objekta preslikom terena na ravnom mediju
2. Problemi s ažuriranjem
3. Skupo i dugotrajno
4. Prostorna analiza na ovaj način je zahtjevnija
5. Prikaz oduzima puno vremena
6. Problem skladištenja
7. Problematično održavanje
8. Složenost pretraživanja podataka
9. Preklapanje
10. Prikupljanje podataka
11. Boje i uzorci koje je teško primijeniti

Iako su se procesi mapiranja u počecima izvršavali isključivo u analognim oblicima, u današnje vrijeme je digitalizacija uzela maha i nad kartografijom. Na taj način riješeni su neki od problema analogne kartografije, ali je stvoren i niz novih problema. Uz primjenu web tehnologija, sa digitalizacijom riješeni su problemi kao što su: problemi s ažuriranjem,

prostorna analiza, spori prikaz, održavanje, složenost pretraživanja, te olakšana izrada tematskih karata (primjena boja i uzoraka na kartama)(Kumar, 2018, p 27).

3.2. Mjerila, Simboli, Projekcije

Svaka karta ima barem 3 osnovna obilježja: mora imati svoje mjerilo, mora biti u određenoj projekciji i posjeduje određenu simboliku. Sve karte jesu preslika stvarnosti na određeni medij, što znači da svaki objekt na karti predstavlja stvarni objekt, a svaki put predstavlja točno taj put na karti. Isto je sa udaljenostima, te se iz tog razloga javila potreba za mjerilima. Mjerilo predstavlja odnos dužine (veličine) između pravog svijeta i dužine na karti. Iako za ljude koji se bave kartografijom razlomci i dijeljenje mjerilima ne predstavljaju problem, laicima se ona mogu činiti kompliciranijima nego što jesu. Mjerilo 1 : 100000 označava da 1 jedinica na karti predstavlja 100000 jedinica u stvarnosti. U slučaju kada bi centimetri bili mjerna jedinica, to bi značilo da 1 centimetar na karti predstavlja 1 kilometar u stvarnom svijetu (1 kilometar = 100000 centimetara). Nadalje, pojmovi koji često mogu uzrokovati zabune jesu „sitno“ i „krupno“ mjerilo. Sitno mjerilo predstavlja karte sa većim mjerilom, dok krupno mjerilo označava karte sa manjim mjerilom, na primjer mjerilo 1 : 1000000 sitnije je nego mjerilo 1 : 1000(Hargitai, 2019, p 113).

Kao što je navedeno, karte predstavljaju stvarni svijet na mediju, a kako bi se to moglo uspješno izvršiti potrebni su simboli. Najčešći simboli na kartama jesu točke, linije i poligoni. Na tematskim kartama često se referencira na legendu na kojoj su navedena značenja simbola. Točkama se na karti mogu označavati na primjer: Geodetske točke ili pak gradovi, ovisno o mjerilu i svrsi karte. Linijama se na kartama najčešće označavaju ceste(ulice), putevi, ograde, rijeke, granice i slično. Poligonima se na kartama označavaju kuće, građevine, gradovi, područja i općenito površine (u ovisnosti o mjerilu i svrsi karte)(Wikipedia).

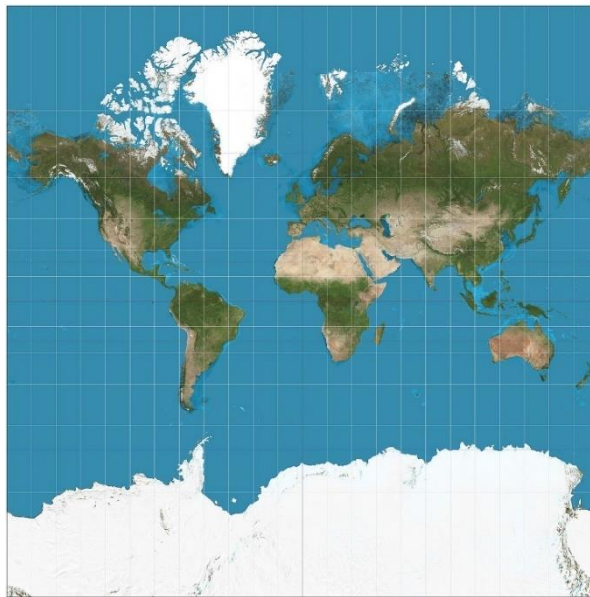
Problem preslikavanja neravne zemljine površine na ravni medij rješavaju projekcije. Svaka od njih to ostvaruje na svoj način, uz određene sličnosti. Generalno, sve matematičke funkcije koje rade transformacije iz koordinata na neku vrstu površina smatraju se projekcijama(Hargitai, 2019, p 84).

S obzirom na probleme sa distorzijom prilikom kartiranja takvih površina, postoje dvije podjele: projekcije s obzirom na očuvanje metričkih svojstava i projekcije s obzirom na precrtavajuću površinu(Bugayevskiy et Snyder, 1995, p 41).

3.2.1. Podjela projekcija sa očuvanjem metričkih svojstava

1. Konformna projekcija

U konformnoj projekciji očuvanje se odnosi na kutove, što znači da ako u stvarnosti postoji kut između dvije ulice iznosi točno 90 stupnjeva, da će taj kut i na karti biti isti. Najpoznatiji primjeri konformne projekcije jesu: Merkatorova projekcija, Lambertova projekcija, stereografna projekcija i Langrangova projekcija(Kennedy et Kopp, 2001, p 12).



Slika 11: Merkatorova projekcija

Izvor:

https://en.wikipedia.org/wiki/Mercator_projection#/media/File:Mercator_projection_Square.JPG

2. Ekvivalentna projekcija

U ekvivalentnoj projekciji površine su očuvane. Primjeri takvih projekcija jesu: Behrmannova, Sinusoidalna, Mollweideova, Bonneova(Kennedy et Kopp, 2001, p 12).



Slika 12: Sinusoidalna projekcija

Izvor: <https://gisgeography.com/equal-area-projection-maps/>

3. Ekvidistantna projekcija

Ekvidistantne projekcije su projekcije koje imaju očuvane dužine, na način da održavaju dužinu jedne ili više linija (dužina) prema svim drugim točkama na karti (Kennedy et Kopp, 2001, p 12).



Equidistant Cylindrical Projection

Maptitude Mapping Software

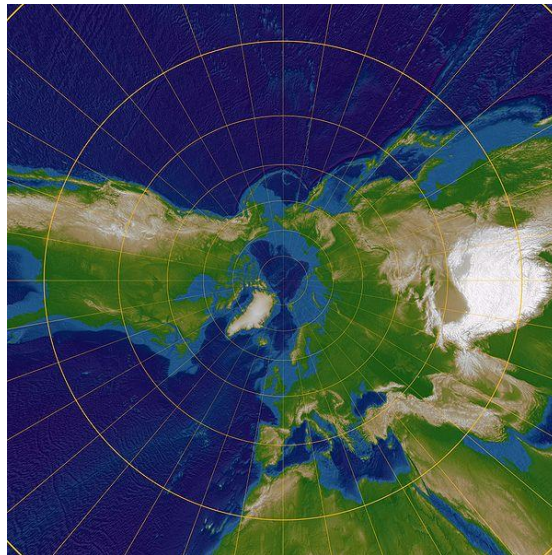
©2013 CALIPER

Slika 13: ekvidistantna cilindrična projekcija

Izvor: <https://www.caliper.com/glossary/what-is-an-equidistant-projection.htm>

4. Gnomonska projekcija

Gnomonska (eng. *Gnomonic*) projekcija je projekcija u kojoj se zakrivljene linije(meridijani i paralele) prikazuju kao ravne linije na projekciji. Smatra se najstarijom projekcijom, a razvio ju je Tales u šestom stoljeću prije Krista(Kennedy et Kopp, 2001, p 18).

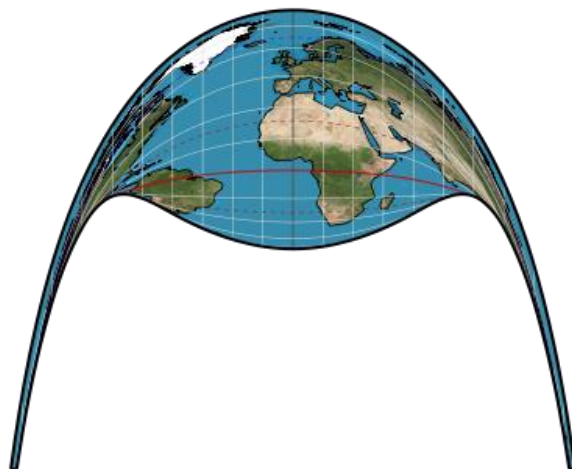


Slika 14: gnomonska projekcija zemlje sa centrom na sjevernom polu.

Izvor: http://wiki.gis.com/wiki/index.php/File:Gnomonic_Projection_Polar.jpg

5. Retroazimutalna projekcija

Na retroazimutalnim projekcijama svaki se smjer(azimut) prikazuje kao i u stvarnosti, dakle od neke točke na karti usmjeren prema centru karte(Bugayevskiy et Snyder, 1995, p 133).



Slika 15: Craig-ova retroazimutalna projekcija sa centrom u Meki

Izvor: <https://www.mapmathematics.com/ProjectionsList.php?Projection=241>

6. Kompromisna projekcija

Kompromisne projekcije nisu bazirane na očuvanju nekih od metričkih svojstava, već da uz pomoć distorzije definiraju što realniji prikaz (Kennedy et Kopp, 2001, p 81).



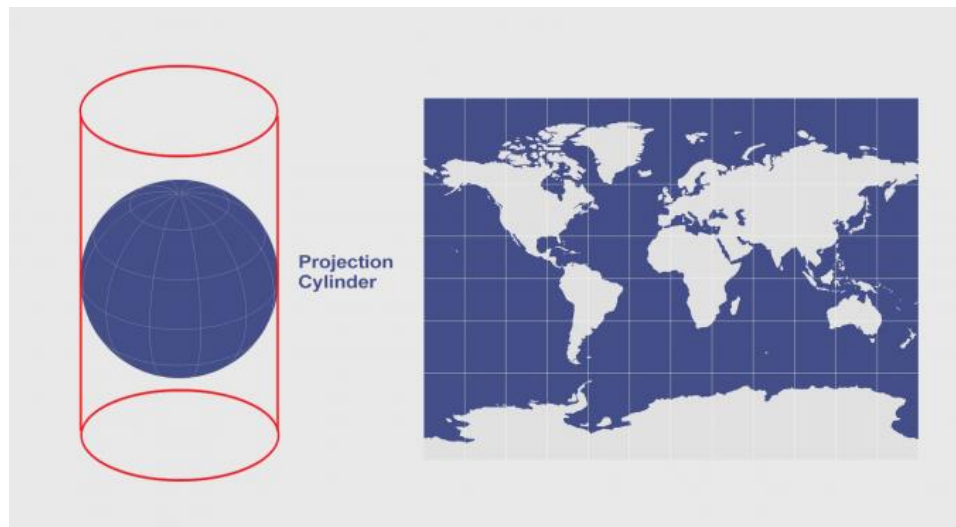
Slika 16: Robinsonova projekcija, primjer kompromisne projekcije

Izvor: <https://geography.wisc.edu/maplibrary/the-robinson-projection/>

3.2.2. Projekcije s obzirom na preslikavajuću površinu

1. Cilindrična projekcija

Cilindrična projekcija je svaka projekcija koja preslikava neko geometrijsko tijelo (u ovom slučaju planet zemlju) na valjak(cilindar). Dakle zemlja se može zamisliti kao da je umotana u valjak(cilindar) na koji se preslikava, te se nakon toga valjak odmotava u ravnu plohu. Na mjestu gdje valjak dodiruje ekvator (može biti i neka druga paralela ili meridijan, ovisno o projekciji) najmanja je distorzija, a raste prema rubovima(Kennedy et Kopp, 2001, p 16).

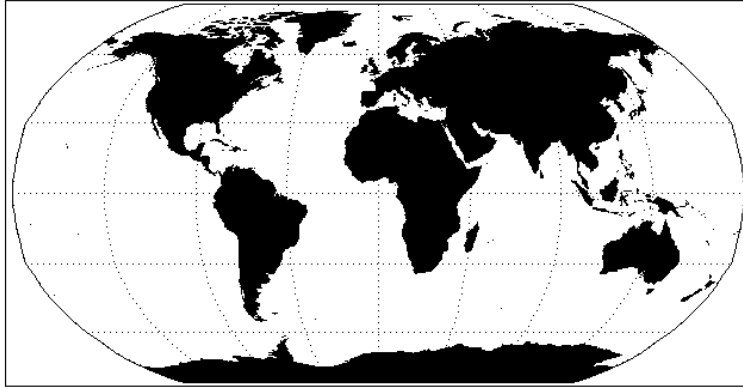


Slika 17: cilindrična projekcija

Izvor: <https://gisgeography.com/cylindrical-projection/>

2. Pseudocilindrična projekcija

Pseudocilindrična projekcija je veoma slična cilindričnoj, a razlika je u tome što su na pseudocilindričnoj meridijani prikazani kao zakrivljene linije. Kao rezultat projekcija nije pravokutna, već je na rubovima zakrivljena. Primjer pseudocilindrične projekcije je Robinsonova projekcija(Kennedy et Kopp, 2001, p 19).



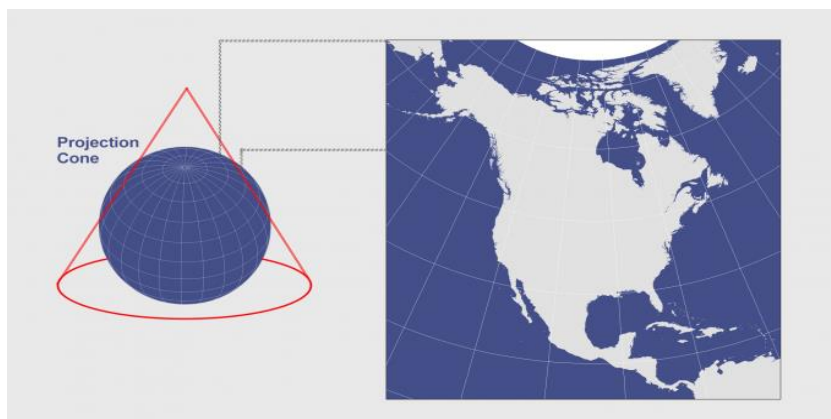
Slika 18: primjer pseudocilindrične projekcije – Robinsonova projekcija

Izvor:

https://climserv.ipsl.polytechnique.fr/documentation/idl_help/Pseudocylindrical_Projections.html

3. Konusna projekcija

Konusna projekcija se može zamisliti tako da se površina zemlje omota u konus(stožac) i precrtava se na njegovu unutrašnjost. Najjednostavnija konusna projekcija dodiruje globus duž jedne linije zemljopisne širine, tangente, koja se naziva standardna paralela. Dužinske linije projicirane na stožastu površinu sastaju se na vrhu, dok se linije zemljopisne širine projiciraju na stožac kao prstenovi. Konus je prerezan duž bilo koje crte zemljopisne dužine i odmotan kako bi se proizvela konačna projekcija. Meridijan nasuprot rezu je središnji meridijan projekcije karte (crvena linija na donjoj slici). Tako precrtana karta ima ravne konvergentne linije zemljopisne dužine s međusobnim rastojanjem. Kao i kod cilindrične projekcije distorzija raste od središta (u ovom slučaju od „standardne paralele“) prema sjeveru i jugu karte, gdje često dolazi do ekstremne distorzije(Department of Geography and Environmental Science, Hunter College).

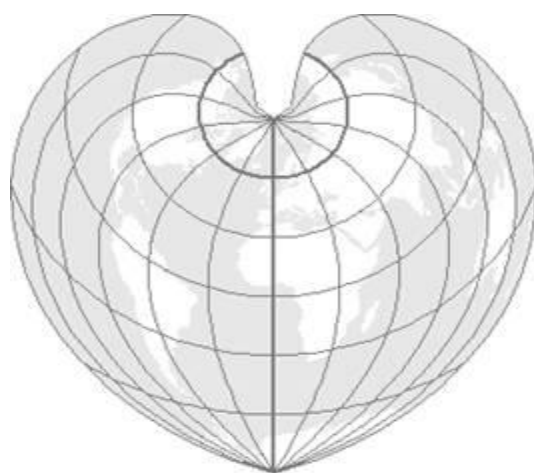


Slika 19: konusna projekcija

Izvor: <https://gisgeography.com/conic-projection-lambert-albers-polyconic/>

4. Pseudokonusna projekcija

Slično kao i sa razlikom između cilindrične i pseudocilindrične projekcije, kod pseudokonusne projekcije meridijani su prikazani kao zakrivljene linije, a paralele su djelomično koncentrične kružnice. Krajnji rezultat projekcije podsjeća na simbol „srca“ (Bugayevskiy et Snyder, 1995, p 395).



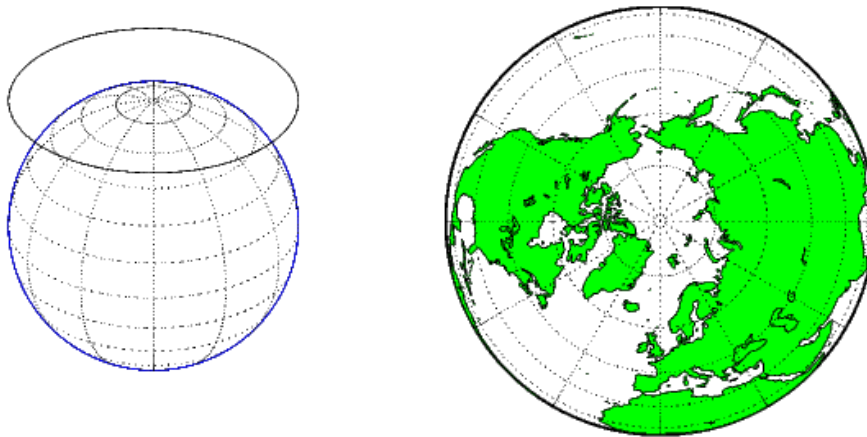
Slika 20: Bonne-ova projekcija, primjer pseudokonusne projekcije.

Izvor:

<http://www.geo.hunter.cuny.edu/~jochen/gtech201/lectures/lec6concepts/Map%20coordinate%20systems/Classifying%20conic%20and%20pseudoconic%20projections.htm>

5. Azimutna projekcija

Kod azimutne projekcije zemlja se precrtava na ravninu. U polarnom aspektu, azimutna projekcija preslikava se na ravninu koja je tangenta na Zemlju na jednom od polova, pri čemu su meridijani projicirani kao ravne linije koje počinju od polova, a paralele su prikazane kao potpune kružnice centrirane na polovima. Azimutne projekcije (osobito pravokutne) mogu imati ekvatorijalne ili kose aspekte. Azimutna projekcija daje percepciju zemljine zakrivljenosti, ali loša strana je da nije pogodna za prikaz cjelokupne zemlje(MathWorks).



Slika 21: Azimutna projekcija. Centar – Sjeverni pol.

Izvor: <https://www.mathworks.com/help/map/the-three-main-families-of-map-projections.html>

3.3. Vrste karata

Iako postoje tisuće vrsta karata, u gruboj podjeli postoje samo dvije vrste: topografske i tematske. Topografske karte sadrže jednako relevantne objekte, te niti jedan od njih nije posebno istaknut(ICS).

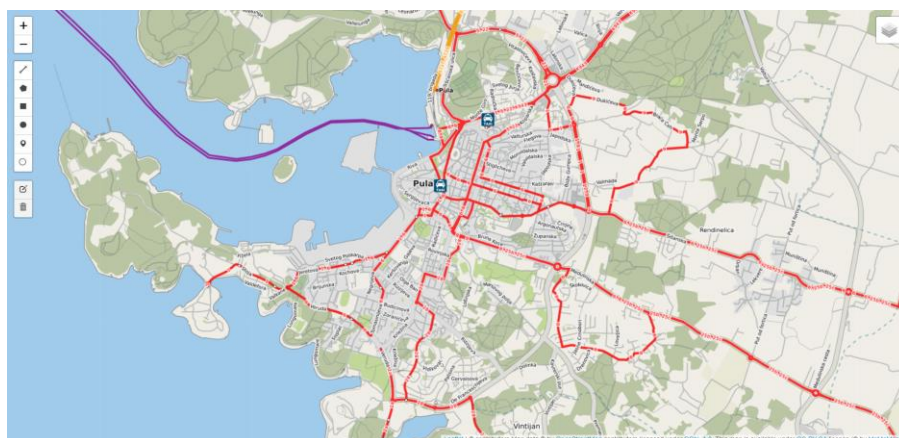
Kao primjer uzeta je karta implementirana u aplikaciji, a može se pronaći i na web stranici: <https://opentopomap.org/>



Slika 22: topografska karta Pule, primjer iz aplikacije

Izvor: <https://opentopomap.org/>

Tematske karte su karte na kojima se neki od elemenata bitno ističu, a postoje brojni primjeri. Među njih spadaju: karte sa raznim istaknutim statističkim značajkama (primjer: karta sa prosječnim prihodima po glavi stanovnika po županijama u Republici Hrvatskoj), karte sa posebno naznačenim objektima (bolnice, crkve i sl.) i karte sa prikazanom vremenskom prognozom i sl. Kao primjer, prikazana je tematska karta implementirana u aplikaciji, sa naglašenim transportnim elementima (autobusne linije, taksi stajališta, brodske linije) (ICSM).



Slika 23: prikaz tematske (transport) karte Pule, primjer iz aplikacije.

Izvor: <http://memomaps.de/en/homepage/>

3.4. Web kartiranje

Web kartiranje (eng. *Web mapping*) je proces korištenja karata nastalih iz nekog od vrsta GIS-a, ali na webu. Web kartiranje obuhvaća i područja šira od same kartografije. Za takvu vrstu kartiranja važni su procesi same izrade i dizajna sustava, kao i pribavljanje i pohrane podataka. Izraz web GIS i web kartiranje su jednim djelom sinonimni. Ključna razlika web kartiranja je mogućnost interakcije korisnika i sustava (klijent-server). Pojava web kartiranja smatra se velikim iskorakom u kartografiji, jer je do nedavno kartografija bila u upotrebi samo od strane nekolicine velikih kompanija i instituta. Razlog tomu jesu kompleksni i skupi zahtjevi (kako hardverski tako i softverski), ali i mali broj stručnjaka na tom području. Web kartiranje je sa sobom donijelo veliki broj setova podataka, uključujući i besplatne podatke napravljene od strane *OpenStreetMap*-a, ali i setova podataka čiji su vlasnici: *Google*, *HERE*, *TomTom*, *Tencent* i drugi. Veći broj djelomično besplatnih alata je također izašao na tržište uključujući alate kao što je *ArcGIS*. Svi ovi događaji rezultirali su povećanim korištenjem i izradom web karata (Scarl et Tochtermann, 2007, p 155).

Prema knjizi „*Settings and needs for web cartography*“ autora M.J. Kraak-a, web karte se ugrubo mogu podijeliti na: Statičke i Dinamičke (Interaktivne). Statičke web karte mogu se samo pregledavati, što znači da ne podržavaju animacije i interaktivnost. Ovakve karte najčešće su napravljene ručno i često su rijetko ažurirane. Tipični grafički formati ovakvih karata su *PNG*, *JPEG*, *GIF*, *TIFF* za rastere, i *SVG*, *PDF*, *SWF* za vektore. Često se koriste karte skenirane sa papira, pri čemu treba paziti na rezoluciju. Web GIS u oblaku mogao bi se pripisati kao dinamička web karta, ukoliko se u njega uključi interaktivnost. Takav način rada razne kompanije sada nude u obliku softvera kao proizvoda (eng. „*Software as a service*“). Cilj toga jest izrada i dijeljenje karta na način da se podaci pohranjuju u oblak.

Osim pojmova koji su već spomenuti potrebno je shvatiti princip rada sljedećih pojmova: Servis web karata (eng. *Web Map Service*) i Popločana web karta (eng. *Tiled web map*)

3.4.1. Web Map Service

„*Web map service*“ je standardni protokol razvijen od strane „*Open Geospatial Consortium*“ još 1999. godine. *Web map service* protokol služi za posluživanje georeferencirane karte preko interneta. Ove fotografije su najčešće generirane sa servera od podataka dobivenih iz GIS baze. WMS definira brojne vrste zahtjeva, ali dva zahtjeva su obvezna: *GetCapabilities* i *GetMap*. *GetCapabilities* vraća parametre o samom WMS-u (naprimjer kompatibilnu verziju WMS-a ili format fotografije karte) i dostupne slojeve (naprimjer koordinatni referentni sustav).

GetMap vraća fotografiju karte, a uključeni parametri su: širina i visina, koordinatni referentni sustav, stil prikazivanja i format fotografije karte. Ostali opcionalni zahtjevi su: „*GetFeatureInfo*“, „*Describe Layer*“ i „*GetLegendGraphic*“.

WMS server najčešće poslužuje karte u formatima: *PNG*, *GIF*, *JPEG* i sl. Dodatno vektorska grafika može biti uključena sa točkama, linijama, tekстом u formatima *SVG* ili *WebCGM(OGC)*.

3.4.2. Tiled web map

Popločana web karta ili popločana karta (eng. *Tiled map*) je prikazana u web pregledniku na način da se neprimjetno preklopi veći broj karata. Svaki manji dio karte, to jest jedna karta ili jedna ploča karte, je individualna datoteka dohvaćena sa servera. Ovakav način rada je najčešći način rada u praksi, te je zamijenio načine rada kao što je servis web karata koji je prikazivao jednu veliku sliku i kontrole u obliku gumba pomoću kojih se pretražuju druga područja. *Google Maps* je bio jedan od prvih velikih servisa koji je počeo sa ovakvim načinom rada, a ostali su slijedili dobar primjer. U današnje vrijeme sve češća je upotreba vektorskih „ploča“, dok su u početku bile korištene rasterske, koje neki sustavi koriste čak i danas. Postoje brojne prednosti ovakvog načina rada sa web kartama. Svaki put kada korisnik popločane web karte zumira ili promijeni lokaciju na karti, dio karata ostaje u prikazu i ne mijenja se, a dio koji je potrebno prikazati dohvaća se sa servera.

Ovakav način rada znatno poboljšava korisničko iskustvo, pogotovo u usporedbi sa korištenjem sustava sa dohvaćanjem jedne velike karte za cijeli ekran. Optimizacija popločanih web karata se može izvršiti čuvanjem bližih dijelova u memoriji i pamćenjem već dohvaćenih karata. Dohvaćanje više manjih karata je već od početka lakši zadatak za izvršiti u usporedbi sa WMS sustavima koji za prikaz koriste jednu veliku sliku za prikaz karte. Iako je većina karata još uvijek u rasterskom formatu (najčešće PNG ili JPG formatima), u današnje vrijeme znatno se razvijaju vektorske karte. U odnosu na rasterske, vektorske popločane karte lakše se mogu stilizirati u korisničkom Internet pregledniku. Definiranje popločanih karti zahtjeva određene konvencije ili standarde, kao što je veličina ploča(karti), razina zumiranja, vrsta projekcija, numeriranje karte i metode kojima se karte dohvaćaju(The Pennsylvania State University).

Većina popločanih web karata slijedi konvenciju *Google Maps-a*:

1. Ploče(karte) su veličine 256 * 256 piksela
2. Na najmanjoj razini zumiranja (0), cijeli svijet je prikazan na jednoj ploči
3. Svaka sljedeća razina zumiranja udvostručuje obje dimenzije, što znači da se svaka ploča „razbija“ na 4 manje ploče. U praksi se obično koriste maksimalno 22 razine zumiranja, za koje se smatra da omogućavaju zadovoljavajuću razinu detalja za širu upotrebu
4. Kao projekcija koristi se projekcija *Web Mercator* (85 stupnjeva je ograničenje za zemljopisnu širinu)(Google)

3.5. Web karte

Iako nije služben, drugi standard je „*OpenStreetMap*“ standard, poznatiji kao „*Slippy Map Tilenames*“ ili „XYZ“, osim već spomenutih karakteristika ima i dodatne:

1. X i Y brojanu shemu
2. Fotografije ploča su u *PNG* formatu
3. Dohvat ploča preko web servera uz pomoć linka(url), kao na primjer „http://map//Z/X/Y.png“ , pri čemu Z predstavlja razinu zumiranja, a X i Y predstavljaju ploču.

Za prikaz popločane karte u Internet pregledniku, potrebno je koristiti preglednik sa podrškom za biblioteke za web karte, koje omogućavaju korisnicima dohvat, prikaz i kretanje po kartama.

3.5.1. OpenStreetMap

Steve Coast osnovao je *OpenStreetMap* 2004. godine, a 2006. godine stvorena je fundacija „*OpenStreetMap Foundation*“ kako bi se potakao rast, razvoj i distribucija besplatnih geoprostornih podataka koje bi svi mogli koristiti i dijeliti. *OpenStreetMap* je stvoren od strane zajednice korisnika *Open Street* karata koji dodaju i održavaju geoprostorne informacije. U održavanje su uključene razne vrste podataka: ceste, putevi, autobusne i željezničke stanice i sl. Podaci *OpenStreetMap*-a uključuju cijeli svijet. *OpenStreetMap* je potpuno besplatan te napravljen uz pomoć volontera od samoga početka i izdan pod „*open-content*“ licencom. Kretanje *OpenStreetMap* kartom je jednostavno, a razina detalja i pokrivenost se može pogledati na području interesa. Tijekom godina razvijanja, projektu su se pridružili razni individualci, vlade, i komercijalne kompanije. Istaknuto je da ovaj projekt nije isključivo web projekt. Razlog tome jest zajednica koja je uključena u ovaj projekt i doprinosi ažurnim informacijama sa terena kako bi poboljšali ovu kartu. Sama zajednica kao i projekt, raste svakog dana te uz njenu pomoć karta postaje sve preciznija. Postoje razni načini kako se priključiti ovom projektu, bilo to mjerenje putova GPS uređajem, korištenjem online „iD“ editora ili JOSM desktop aplikacijom kreiranom isključivo za doradu karte. *OpenStreetMap* pogonjen je softverom otvorenog koda, od njegovog sučelja „klizave karte“ koja omogućava kretanje kartom i zumiranje, do podataka koje se nalaze na web servisu(*OpenStreetMap*).

3.5.2. MapBox

Mapbox osnovan je 2010. godine, a cilj mu je bio rad sa neprofitnim i humanitarnim organizacije uključujući geoprostorne podatke i analize. *Mapbox* se u međuvremenu razvio, te je svoje karte, alate i podatke počeo pružati kompanijama poput: *Facebook*-a i *Snapchata*-a. Osim velikih kompanija *Mapbox* na umu ima i manje poduzetnike. Iako *Mapbox* servisi više nisu potpuno besplatni, i dalje rade na softveru otvorenog koda. Slično kao i kod svoje konkurencije, *Mapbox* u ponudi ima set alata i značajki za integraciju karti u bilo koju mobilnu ili web aplikaciju. Također nude i alat imena „*Mapbox Studio*“ u kojemu je moguće kreirati vlastiti dizajn i uklopiti karte u vlastite proizvode. Bilo da je potrebna vizualizacija podataka, navigacija ili lociranje, *Mapbox* nudi rješenja. *Mapbox* se spominje upravo u ovom dijelu jer ima vlastiti servis za pločaste karte: „*Mapbox Tiling Service*“ ili skraćeno MTS. MTS nudi pretvorbu geoprostornih podataka u vektorske ploče sa redovitim ažuriranjima, a ponudi je i visoko propusna mreža sa niskom latencijom. MTS ima vlastitu infrastrukturu i integrira se u aplikacije uz pomoć svojih API-a ili SDK-ova. Pokreće se na *MapBox*-ovoj globalnoj platformi što omogućuje brzi pristup podacima. Za korištenje MTS potrebna su dva inputa za generiranje pločastih karti: geoprostorni podaci(izvor pločastih karti) i JSON konfiguracijski dokument kojim se specificiraju opcije uz pomoću kojih se izvor pretvara u karte(*MapBox*).

3.5.3. Google Maps

Google Maps Platforma pokriva 99% cijelog svijeta, a koristi je preko milijardu korisnika na mjesečnoj bazi. Već neko vrijeme *Google maps* je industrijski standard za integraciju karti, a koriste ga i kompanije poput *Bolta*, *Ubera* i mnogih drugih velikih i malih kompanija. *Google maps* isporučuje ogromne količine detaljnih informacija sa ažuriranjima u pravom vremenu uz pomoć satelita, Android uređaja, „*Street view*“ automobila i lokalnih doprinosa. Osim karti(*Google Maps*), dostupni su i servisi poput *Google Routes* i *Google Places*, uz pomoć kojih se pokrivaju skoro sve potrebe industrije(*Google*).

3.6. Razvijanje karti pomoću Web tehnologija

Razvijanje karti pomoću web tehnologija podrazumijeva upotrebu web alata za razvijanje, prikaz i interakciju sa kartama i podacima koji se prikazuju na karti. Ovakav način razvijanja karata znatno se razlikuje od tradicionalne kartografije. Trenutno, geoprostorni podaci i tehnike se koriste u mnogim područjima: logistika, prodaja, planiranje, medicina, u raznim granama menadžmenta, u vojne svrhe i sl. Postoje mnoga gotova rješenja od kojih su neka komercijalna, dok su druga „*open-source*“ te su nadograđivane od strane zajednice sa najnovijim informacijama. Sva rješenja koriste slične sustave i projekcije, ali ipak postoje sitne razlike među njima (Tambassi, 2019, p 258).

S obzirom da ovaj rad obrađuje i izradu aplikacije sa elementima GIS-a, neki od alata i tehnologija obrađeni su u narednim poglavljima. Kao glavna tehnologija u ovom radu izabrana je *JavaScript/TypeScript* sa obzirom da prednjači u razvijanju web aplikacija. Korištene su biblioteke (eng. *Libraries*) *React* i *Leaflet*, a postoje i druga slična rješenja koja služe za izradu karata, poput: *ArcGIS*, *OpenLayers*, *Google Map*.

3.6.1. Leaflet

Leaflet za sebe tvrdi da je „vodeća *JavaScript* biblioteka otvorenog koda za interaktivne karte prilagođene mobilnim uređajima. S težinom od samo 39 KB JS-a, ima sve značajke mapiranja koje su većini programera ikada potrebne.“

Leaflet osim toga nudi jednostavnost, performanse i iskoristivost, a prednost mu je i što efikasno radi na svim popularnijim mobilnim i desktop platformama. S obzirom da se radi o *JavaScript* biblioteci, *Leaflet* može biti proširen sa raznim dodacima, a njegove funkcionalnosti mogu se pronaći na službenoj dokumentaciji. Autor *Leaflet*-a je Vladimir Agafonkin. Ipak, bitno je naglasiti da je *Leaflet* projekt otvorenog koda, te da se njegovom razvoju može pridružiti bilo tko. *Leaflet* ima manji broj značajki u odnosu na konkurenciju, ali je fokusiran na to da osnovne značajke rade savršeno.

Osnovne značajke *Leafleta*:

1. Slojevi(eng. *Layers*) koji rade odmah nakon instalacije: Pločasti slojevi(eng. *Tile Layers*), „*Web Map service*“(WMS), markeri, skočni prozori, vektorski slojevi(linije, poligoni, krugovi, pravokutnici), preklapanje fotografija i *GeoJSON*
2. Interaktivne značajke za olakšavanje rada: povlačenje sa inercijom, zumiranje putem kotačića na mišu, zumiranje s dva prsta na mobitelu, dvostruki klik za zumiranje, zumiranje na područje, događaji na klik miša(eng. *Click events*) i prelazak miša, povlačenje oznaka i sl.
3. Vizualne značajke: animirano zumiranje, animacije skočnih prozora, posebni dizajn za oznake, skočne prozore i kontrole na karti
4. Značajke za prilagođavanje: CSS stiliziranje za skočne prozore i kontrole za jednostavno mijenjanje stilova, oznake s HTML i foto osnovom te prilagodba projekcija
5. Značajke performansi: ubrzanje za mobilne aplikacije, korištenje CSS značajki za glatke animacije, pametno iscrtavanje slojeva (poligoni i linije), modularni sustav (koriste se samo značajke u upotrebi) i eliminacija odgode na dodir (mobilni uređaji)
6. Kontrole na karti: gumbi za zumiranje, atribucija, izmjenjivač slojeva i mjerilo.
7. Podržani preglednici za desktop: *Chrome, Firefox, Safari, Opera, Internet Explorer, Edge*, i za mobilne uređaje: *Safari, Android browser, Chrome, Firefox i Internet Explorer(Leaflet)*.

3.6.2. ArcGIS

ArcGIS je GIS razvijen od strane Esri-a, a postoji više inačica samog sustava. S obzirom na tematiku web kartiranja, fokus je na verziji *ArcGIS Online*. *ArcGIS Online* je softversko rješenje u cloudu za izradu i dijeljenje interaktivnih web karti. Također s obzirom da je fokus rada *JavaScript/TypeScript* tehnologija, naročito je zanimljiv *ArcGIS API* za *JavaScript*. „*ArcGIS API* za *JavaScript*“ jednostavan je način za ugradnju karata i zadataka u web aplikacije. Ove karte možete nabaviti s *ArcGIS Online*-a, vlastitog *ArcGIS* servera ili drugih poslužitelja.

Osnovne značajke *ArcGIS*-a:

1. Podrška za popularne Internet preglednike: *Chrome, Firefox, Safari, Edge*
2. *ArcGIS Online* integracija: korištenje izrađenih karata
3. Optimizacija za mobilne uređaje: API izrađen posebno za mobilne uređaje, uređivanje podataka korištenjem mobilnih uređaja, skočni prozori dizajnirani za mobilne uređaje, podrška za *Android* i *iOS*
4. Podrška za bilo koji tip slojeva: servis za popločane karte (eng. *Tile Map Service*), servis za dinamičke karte (eng. *Dynamic map service*), slojevi značajki, grafika (*canvas, SVG*), *KML, OpenStreetMap, Bing, WMS*
5. *Widgeti* prevedeni na više od 25 jezika
6. Kompatibilnost sa *jQuery JavaScript* bibliotekom (ESRI)

3.6.3. OpenLayers

OpenLayers olakšava izradu i postavljanje karata na bilo koju web lokaciju. Moguće je prikazivati razne vrste slojeva kao što su TMS, DMS, vektorske podatke ili pak markere iz raznih izvora. Razvijen je za uporabu svih vrsta geografskih podataka. U potpunosti je besplatan, a objavljen je pod BSD licencom (poznatom kao *FreeBSD*). Također je softver otvorenog koda, razvijen za korištenje uz druge web tehnologije.

Osnovne značajke *OpenLayers*-a:

1. Popločani sloj karata (eng. *Tiled Layers*): Slično kao i kod *Leafleta*, *OpenLayersu* se mogu nadodati razne karte: *OpenStreetMap, Bing* karte, *MapBox* karte i bilo koji drugi servisi koji nude takav tip karata.
2. Vektorski slojevi podržavaju podatke raznih tipova kao što su: *GeoJSON, TopoJSON, KML, GML, Mapbox* vektorske ploče i još mnoge druge formate.
3. Optimizacija za mobilne uređaje: svi elementi iz HTML, *Canvas 2D, WebGL*, te nadogradnja prilagođenih komponenti prema potrebi.

4. Jednostavno prilagođavanje i nadograđivanje: karte i kontrole na kartama mogu se uređivati CSS-om. Također moguće je povezivanje sa nekim od API-a ili drugih biblioteka sa dodatnim funkcionalnostima(OpenLayers).

3.6.4. Google Maps JavaScript API

Osim za karte, *Google Maps* Platforma podržava i biblioteke za rute i mjesta(Google). *Google Maps* ima opširnu dokumentaciju, ali za korištenje *Google Maps* alata potrebno je pridobiti *API Key* uz pomoć kojega se mogu početi koristiti *Google Maps* alati. Dio alata besplatan je za korištenje, dok je za naprednije alate potrebna pretplata na komercijalnu verziju. *Google Maps JavaScript API* se koristi za izradu vlastitog sadržaja i simbolike, odnosno za prikaz karti na web stranicama i mobilnim uređajima. Nadalje, postoje četiri osnovne vrste karata: „*roadmap*“, „*satellite*“, „*hybrid*“ i „*terrain*“. Sve osnovne vrste karata mogu se nadograditi i prilagoditi dodatnim slojevima, stilovima, kontrolama događaja, servisima i dodatnim bibliotekama. Iako je *Google maps* robusniji sustav od prethodno navedenih (izuzetak *ArcGIS*), u ovom poglavlju uspoređene su samo biblioteke vezane za izradu web karata(Google).

4. Razvoj aplikacije

Kao aplikacijski dio rada odrađena su 2 zadatka: inicijalizacija karte sa alatima za crtanje, i geovizualizacija svih centara u Republici Hrvatskoj u kojima je moguće testirati se na CoVid-19.

Kao tehnologije za obradu ovih zadataka odabrane su sljedeće tehnologije: HTML, CSS, *JavaScript/TypeScript*, *React*, *Leaflet(React-Leaflet)*, *React-Leaflet-Draw*. Nadalje, korišteni su alati: *Visual studio code*, *Notepad++*, *Git*, te Internet preglednici *Chrome* i *Firefox*.

Aplikacija je inicijalizirana naredbom: „*npx create-react-app ime-aplikacije --template typescript*“, te su se uz pomoć te naredbe generirale sve datoteke potrebne za rad sa *React* aplikacijama i *Git* repozitorij.

React-leaflet i *React-leaflet-draw* biblioteke instalirane su putem naredbi i to sljedećim redoslijedom: „*npm install react react-dom leaflet*“, „*npm install react-leaflet*“ i „*npm i react-leaflet-draw*“. *React Leaflet* kao zadanu kartu koristi *OpenStreetMap Classic*, a postoji mogućnost implementacije i drugih karti. Najčešći način dohvata karti je preko web tile servera.

Za projekciju *OpenStreetMap* koristi „EPSG:3857“, što je *Merkatorova* projekcija, ali na sferi(kugli)(*OpenStreetMap*).

Projekt je osim sastavnih datoteka *React* aplikacije proširen sa 2 glavne komponente: „*MapComponent.tsx*“ i „*MarkerComponent.tsx*“. Kroz „*MapComponent*“ prikazana je karta, ali se u njoj poziva i komponenta „*MarkerComponent*“ pomoću koje se zatim prikazuju označene lokacije.

4.1. Inicijalizacija karte (*React-leaflet*)

Uz pomoć *React Leaflet* biblioteke omogućena je veza između *React*-a i *Leaflet*-a. *React Leaflet* ne služi kao zamjena *Leafletu*, već uz pomoć njega apstrakcijom slojeva *Leafleta* omogućuje korištenje tih slojeva kao *React* komponente. Iz tog razloga postoje razlike u ponašanju klasične *React* aplikacije, to jest njenih komponenti. Najveće su razlike u iscrtavanju „dokument objekt modela“ (eng. *Document object model*). *React* ne prikazuje slojeve *Leafleta* u svom dokument objekt modelu, već samo iscrtava `<div>` element pomoću kojeg se prikazuje *Leaflet* karta, sa njezinim značajkama (*React-Leaflet*).

Za inicijalizaciju karte unutar „*MapComponent*“ komponente poziva se komponenta iz biblioteke *React-Leaflet* koja je prethodno uvezena.

```
<MapContainer  
  center={position}  
  zoom={zoom}  
  scrollWheelZoom={true}  
>
```

Slika 24: „*MapContainer*“ komponenta, programski kod aplikacije

Kao parametri prethodno su postavljeni: inicijalna „*zoom*“ razina, centar (početna točka) i omogućeno je zumiranje kotačićem miša. Za prikaz karte potrebno je postaviti „*TileLayer*“ komponentu iz *Leaflet*-a, unutar komponente „*MapContainer*“. Kao inicijalnu kartu moguće je postaviti razne karte. U aplikaciju je uključeno 9 različitih karata koje su besplatne za korištenje, a svaka od njih se dohvaća preko njihovog servera. „*LayersControl*“ komponentom prikazuje se izbornik za mijenjanje karata, a putem „*LayersControl.BaseLayer*“ dodane su karte sa komponentama „*TileLayer*“, kao u inicijalnom prikazu. Komponenta „*TileLayer*“ sadrži dodatne parametre („*props*“) za pripisivanje autorskih prava autorima karata, te „*url*“ za dohvat karti sa servera. U CSS datoteci je definirana visina i širina na 100% za „*MapContainer*“ komponente preko CSS klase „*leaflet-container*“.

```
LayersControl position="topright" JosipTadic, 3 weeks ago • enabled changing tile layers
<LayersControl.BaseLayer checked name="OpenStreetMap Classic">
  <TileLayer
    attribution="<a href='http://osm.org/copyright'>OpenStreetMap</a> contributors"
    url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png"
  />
  <LayersControl.BaseLayer name="OpenTopoMap">
    <TileLayer
      attribution='&copy; Kartendaten: © <a href="http://osm.org/copyright"> OpenStreetMap</a>-Mitwirkende, SRTM | Kartendarstellung'
      url="https://c.tile.opentopomap.org/{z}/{x}/{y}.png"
    />
  </LayersControl.BaseLayer>
</LayersControl.BaseLayer>
<LayersControl.BaseLayer name="Humanitarian map style">
  <TileLayer
    attribution='&copy; <a href="http://osm.org/copyright">OpenStreetMap</a> contributors | base map by <a href="http://www.openstre'
    url="http://b.tile.openstreetmap.fr/hot/{z}/{x}/{y}.png"
  />
</LayersControl.BaseLayer>
<LayersControl.BaseLayer name="wmflabs Hillshading">
  <TileLayer
    attribution='&copy; <a href="http://osm.org/copyright">OpenStreetMap</a> contributors, by <a href="https://outreachdashboard.wmf'
    url="http://tiles.wmflabs.org/hillshading/{z}/{x}/{y}.png"
  />
</LayersControl.BaseLayer>
```

Slika 25: „LayersControl“ i „TileLayer“ komponente za dohvat i prikaz karata, programski kod aplikacije

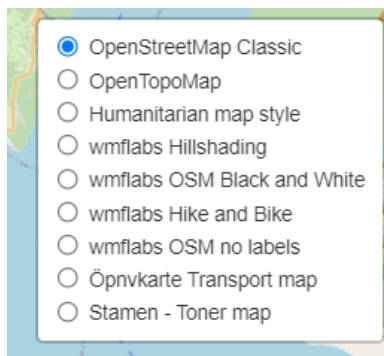
Ikona za izmjenu karata nalazi se u gornjem desnom kutu ekrana.



Slika 26: „LayersControl“ ikona za promjenu karata

Izvor: <https://react-leaflet.js.org/docs/example-layers-control/>

Na prijelaz mišem otvara se izbornik za promjenu.

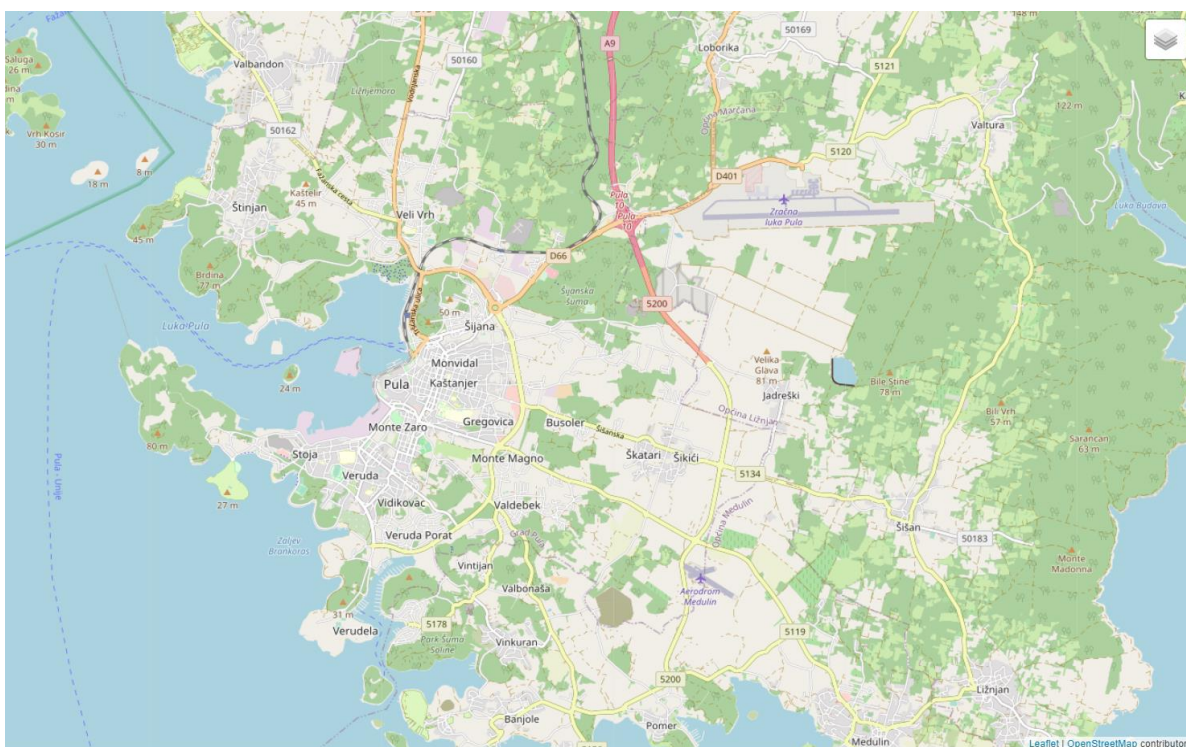


Slika 27: izbornik za mijenjanje karata, primjer iz aplikacije

Izvor: <https://react-leaflet.js.org/docs/example-layers-control/>

Karte uključene u aplikaciju su sljedeće:

1. *OpenStreetMap Classic* (opisan u prethodnom poglavlju)



Slika 28: *OpenStreetMap classic*, grad Pula i okolica, primjer iz aplikacije

Izvor: (url) <https://s.tile.openstreetmap.org/{z}/{x}/{y}.png>

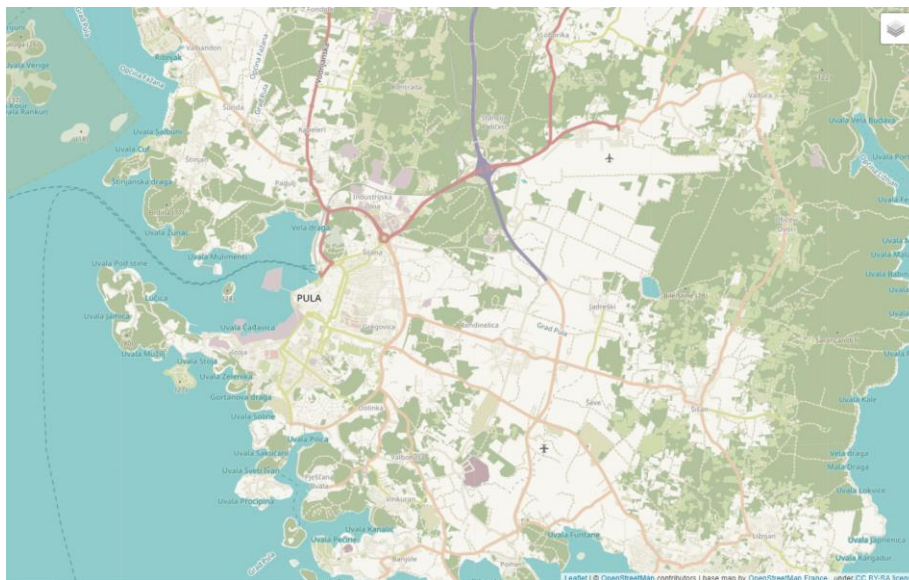
2. *OpenTopoMap*: topografska karta napravljena na osnovu *OpenStreetMap* karte, sa dodatkom visinskih podataka(*OpenStreetMap*).



Slika 29: *OpenTopoMap*, grad Pula i okolica, primjer iz aplikacije

Izvor: (url) <https://c.tile.opentopomap.org/{z}/{x}/{y}.png>

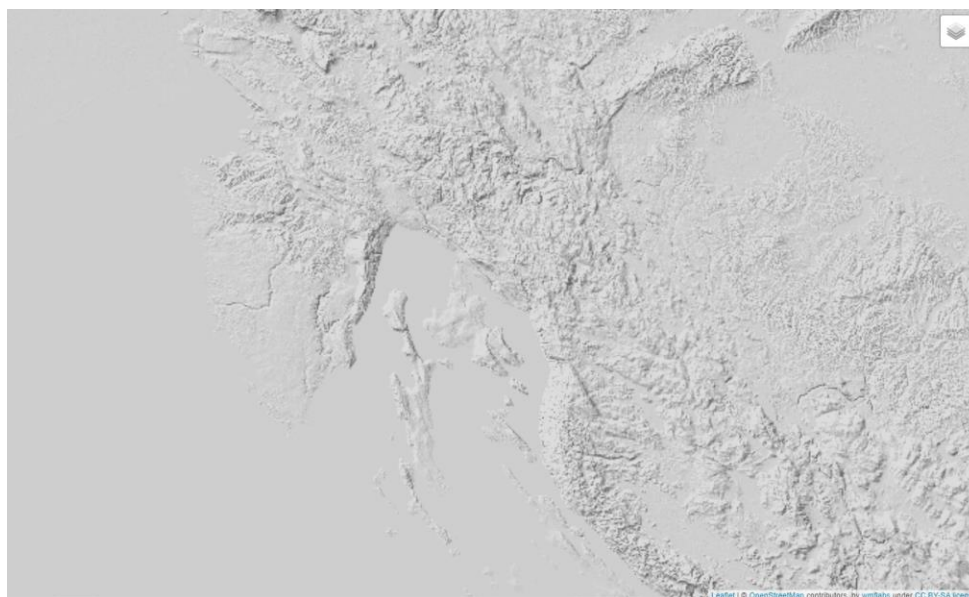
3. *Humanitarian map style*: tematska karta fokusirana na resurse koje su korisni u hitnim slučajevima, kao npr: resursi vode, rasvjeta i javne zgrade. Boje su svjetlijih nijansi kako bi se moglo lakše označavati na ispisanom karti. Često se njima koriste humanitarne organizacije u kriznim situacijama(*OpenStreetMap*).



Slika 30: Humanitarian Map Style, grad Pula i okolica, primjer iz aplikacije

Izvor: (url) <http://b.tile.openstreetmap.fr/hot/{z}/{x}/{y}.png>

4. *wmflabs Hillshading*: crno bijela reljefna karta(OpenStreetMap).



Slika 31: *wmflabs Hillshading* map, Istra, Primorje i Kvarner, primjer iz aplikacije

Izvor: url: <http://tiles.wmflabs.org/hillshading/{z}/{x}/{y}.png>

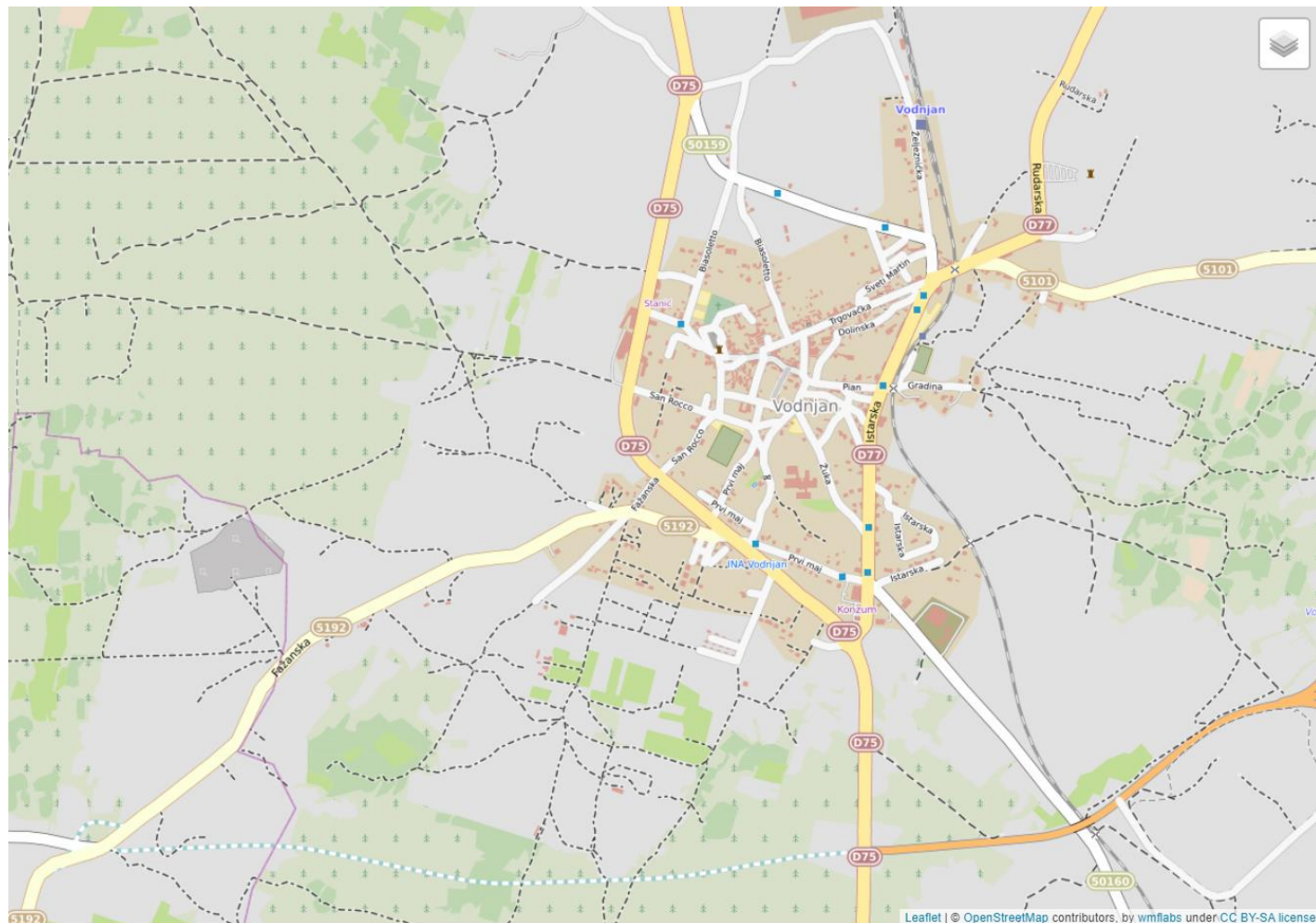
5. *wmflabs OSM Black and White*: varijacija *OpenStreetMap* karte koja koristi samo nijanse crno bijele boje(*OpenStreetMap*).



Slika 32: *OSM Black and White*, grad Pula i okolica, primjer iz aplikacije

Izvor: (url) <https://tiles.wmflabs.org/bw-mapnik/{z}/{x}/{y}.png>

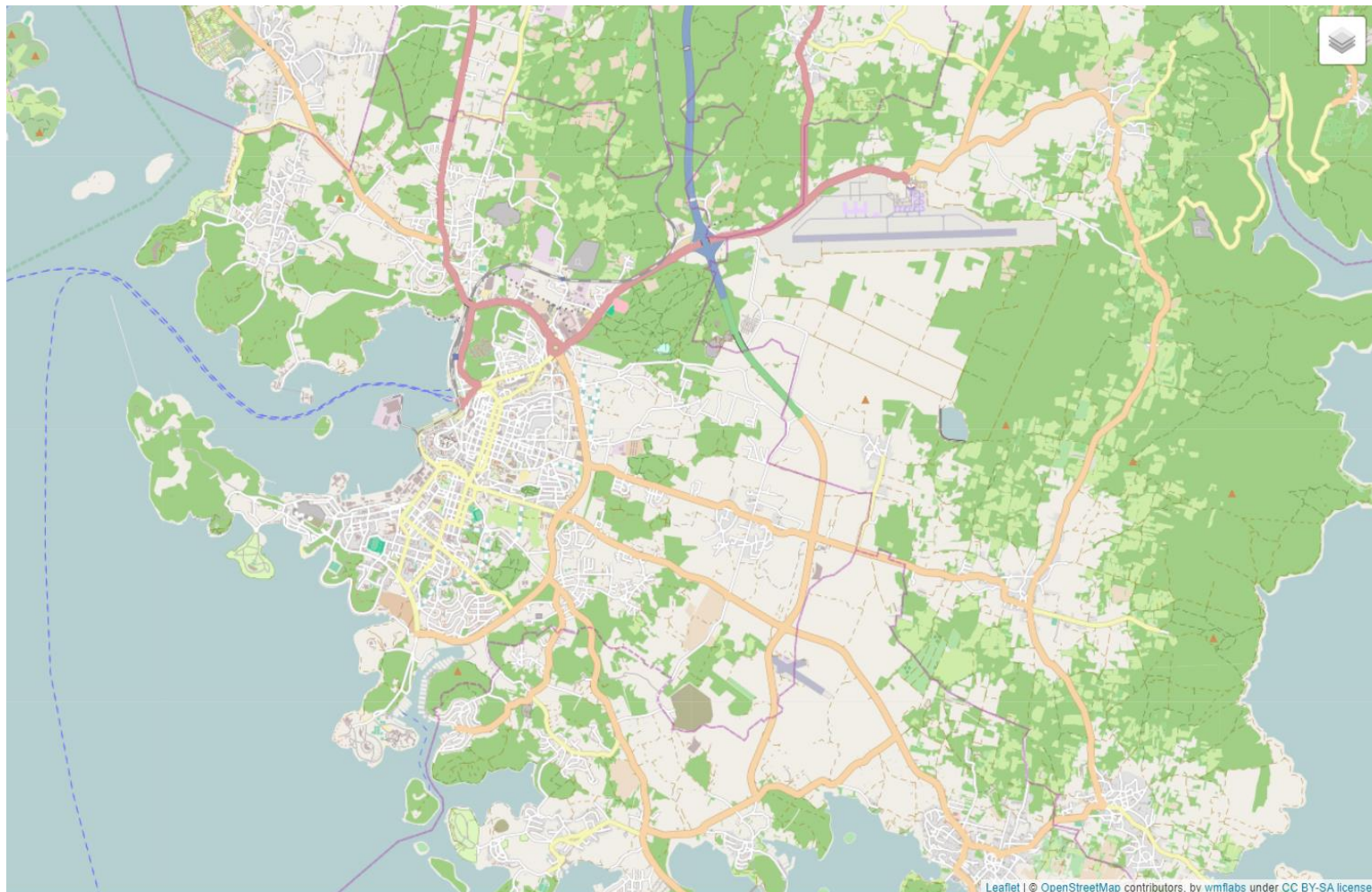
6. *wmflabs Hike and Bike*: karta prilagođena za pješake i bicikliste(OpenStreetMap).



Slika 33: *wmflabs Hike and Bike*, grad Vodnjan, primjer iz aplikacije

Izvor: (url) <https://tiles.wmflabs.org/hikebike/{z}/{x}/{y}.png>

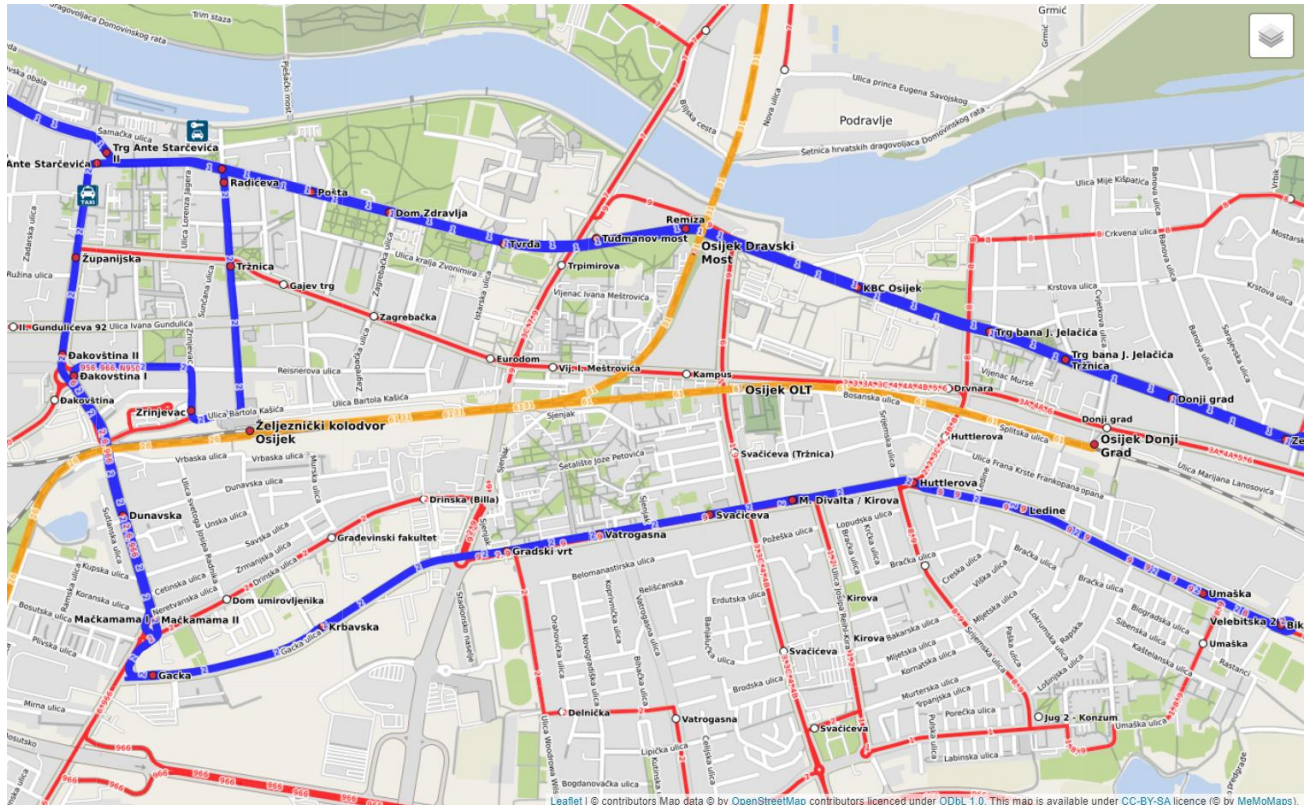
7. Wmflabs OSM no labels: varijacija OpenStreetMap karte na kojima nema natpisa(OpenStreetMap).



Slika 34: wmflabs OSM no labels, grad Pula, primjer iz aplikacije

Izvor: (url) <https://tiles.wmflabs.org/osm-no-labels/{z}/{x}/{y}.png>

8. *Öpnavkarte Transport map*: karta bazirana na *OpenStreetMap* bazi podataka, a naglasak je na prikazu javnog prijevoza (MeMoMaps).



Slika 35: *Öpnavkarte Transport map*, grad Osijek, primjer iz aplikacije

Izvor: (url) <https://tiles.wmflabs.org/osm-no-labels/{z}/{x}/{y}.png>

9. *Stamen - Toner map*: visoko kontrastna crno bijela karta. Pogodna je za naznačavanje podataka, pregled obalnih zona i rubova vodenih površina. *Stamen Toner* karta se također bazira na *OpenStreetMap*-u, a nudi i druge vrste karti kao što su „*Terrain*“ i „*Watercolor*“ karta(Stamen).



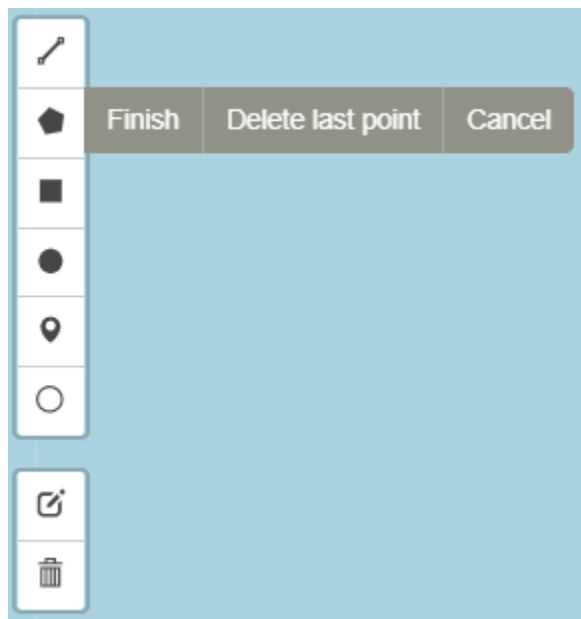
Slika 36: *Stamen Toner map*, grad Venecija, Italija, primjer iz aplikacije

Izvor: (url) <https://stamen-tiles.a.ssl.fastly.net/toner/{z}/{x}/{y}.png>

4.2. Inicijalizacija alata za crtanje *React-leaflet-draw*

React Leaflet draw je komponenta izrađena na osnovu *React Leafleta*, a služi za implementaciju *Leaflet draw* značajki.

Leaflet draw je biblioteka uz pomoć koje se na karti mogu iscrtavati različiti vektorski elementi, poput: poligona, linija, kvadrata, krugova. Osim toga, mogu se i označavati točke sa kružnom ili regularnom oznakom(*marker*). Svaki od ovih elemenata naknadno se može izmijeniti ili obrisati(*React-Leaflet-Draw*).



Slika 37: „*React Leaflet Draw*“ izbornik, primjer iz aplikacije

Izvor: <https://www.npmjs.com/package/react-leaflet-draw>

Za korištenje navedene značajke potrebno je „*EditControl*“ komponentu smjestiti unutar „*FeatureGroup*“ komponente. „*EditControl*“ komponenta prima 2 parametra(*props*): *position* (za poziciju izbornika), te *draw* (onemogućavanje pojedinih elemenata).

```
<FeatureGroup>
  | <EditControl position="topleft" draw={{}} />
</FeatureGroup>
```

Slika 38: „EditControl“ (React Leaflet Draw) komponenta, programski kod aplikacije

Za iscrtavanje elemenata potrebno je kliknuti na željeni element i postaviti početnu točku na karti. Za izmjenu i brisanje potrebno je kliknuti na odgovarajuće ikonice, kojima se odabire element koji se želi izmijeniti ili obrisati. Iscrtavanje linija i poligona(površina) potrebno je potvrditi u izborniku, jer se uvijek može dodati još jedna točka. Također pri operaciji izmjene ili brisanja potrebno je potvrditi ili odbaciti promjene(Leaflet).



Slika 39: iscrtani vektorski elementi, „React Leaflet draw“, primjer iz aplikacije

Izvor karte: <https://s.tile.openstreetmap.org/{z}/{x}/{y}.png>

testnih centara. Iako u tablici postoje prazne ćelije, kroz aplikaciju se naknadno rukuje njima. Na osnovu tablice kreirana je .json datoteka iz kojeg je naknadno formirano polje objekata koji služe za prikazivanje podataka iz tablice.

```
{ You, 3 days ago • added CoVid testing centers on map, fixed di
  id: "23",
  header: "KBC OSIJEK",
  address: "Ul. Josipa Huttlera 4, 31 000 Osijek",
  contact: "031 511 050 ",
  price: "698,21 Kn",
  openHours: "Od ponedjeljka do subote od 07:30 - 10:30",
  driveIn: "",
  waitTime: "do 24 sata",
  translate: "DA - NE",
  lon: 18.7145154,
  lat: 45.5577195,
},
```

Slika 40: objekt sa podacima o centru za testiranje KBC Osijek tipa JSON, programski kod aplikacije

Kako bi podaci strukturno pratili aplikaciju pomoću *TypeScript interface*-a definirani su tipovi podataka. Tipovi podataka koji su označeni znakom „?” označavaju neobavezne podatke. Uzrok tome su nepotpuni podaci, koji zahtijevaju posebno rukovanje. U slučaju kada bi oni obavezni, a njihove vrijednosti ne dodijeljene kasnije u programskom kodu, *TypeScript*-ov interpreter bi onemogućio pokretanje aplikacije.

```
interface MarkerComponents {  
  id: string;  
  header: string;  
  address: string;  
  contact?: string;  
  price?: string;  
  openHours?: string;  
  driveIn?: string;  
  waitTime?: string;  
  translate?: string;  
  lon: number;  
  lat: number;  
}
```

josip, 2 months ago • added

Slika 41: „MarkerComponents“ interface, definicija tipova, programski kod aplikacije

Kako bi podaci bili prikazani u aplikaciji definirana je varijabla „icons“, koja je polje tipa „MarkerComponents“. U varijablu (polje) „icons“ su spremljeni svi preuzeti podaci.

```
const icons: MarkerComponents[] = [
  {
    id: "1",
    header: "HRVATSKI ZAVOD ZA JAVNO ZDRAVSTVO",
    address: "Rockefellerova 2, 10 000 Zagreb",
    contact: "covid@hzjz.hr",
    price: "150,00 kn(390,00 kn)",
    openHours:
      "Svaki radni dan od 07:30 - 09:30 sati te subotom od 08:00 - 10:00 sati",
    driveIn: "DA",
    waitTime: "24 - 48 sati",
    translate: "DA - DA",
    lon: 15.9809638,
    lat: 45.8291116,
  },
  {
    id: "2",
    header: "NZJZ DR. ANDRIJA ŠTAMPAR",
    address: "Mirogojska cesta 16,10 000 Zagreb",
    contact: "covidtest@stampar.hr",
    price: "500,00 Kn",
    openHours:
      "Od ponedjeljka do petka od 8 do 14 h, subotom i nedjeljom od 8 do 12 sati",
    driveIn: "DA",
    waitTime: "24 sata",
    translate: "DA - DA",
    lon: 15.9806294,
    lat: 45.8315145,
  },
  {
    id: "3",
```

Slika 42: primjeri JSON polja objekata „MarkerComponents“, programski kod aplikacije

Za prikaz CoVid centara korištena je *React Leaflet* komponenta <Marker />, a za prikaz atributnih podataka korištena je druga *React Leaflet* komponenta <Popup />. Uz pomoć koje je stvoren skočni prozor u kojemu su prikazani navedeni podaci. U komponenti „Marker“, lokacije su proslijeđene kao varijabla s 2 polja (širina i dužina). Kao parametri komponente ikone(*markera*) proslijeđen je objekt „pinIcon“, koji sadrži definicije o ikoni. Ključ po kojem se „mapiraju“ komponente je „id“, a kao naslov skočnog prozora definiran je „header“. U objektu „pinIcon“ definirana su sidra(*anchor*), od kojih jedno

odmiče skočni prozor 15 piksela od ikone („*popupAnchor*“) te drugo („*iconAnchor*“) kojom se ikona centrira u ishodište(0,0). Uz pomoć „*iconSize*“ vrijednosti definirana je veličina mjesta za ikonu. Sa „*className*“ „*marker-component-icon*“, dodijeljeni su CSS stilovi.

```
const pinIcon: L.DivIcon = L.divIcon({
  className: "marker-component-icon",
  popupAnchor: [15, 0],
  iconAnchor: [0, 0],
  iconSize: [25, 25],
});
```

Slika 43: Objekt „*pinIcon*“, definicije markera.

CSS datoteka uvezena je u komponentu, sa sljedećim stilovima: „*border-radius*“, „*box-shadow*“, „*display*“, „*width*“, „*height*“, „*content*“ i „*font-size*“.

```
.marker-component-icon {
  border-radius: 50%;
  box-shadow: 0px 0px 10px  black;
}

.marker-component-icon:after {
  display: block;
  width: 20px;
  height: 20px;
  content: "📍";
  font-size: 20px;
}
```

Slika 44: CSS stilovi za ikone, programski kod aplikacije

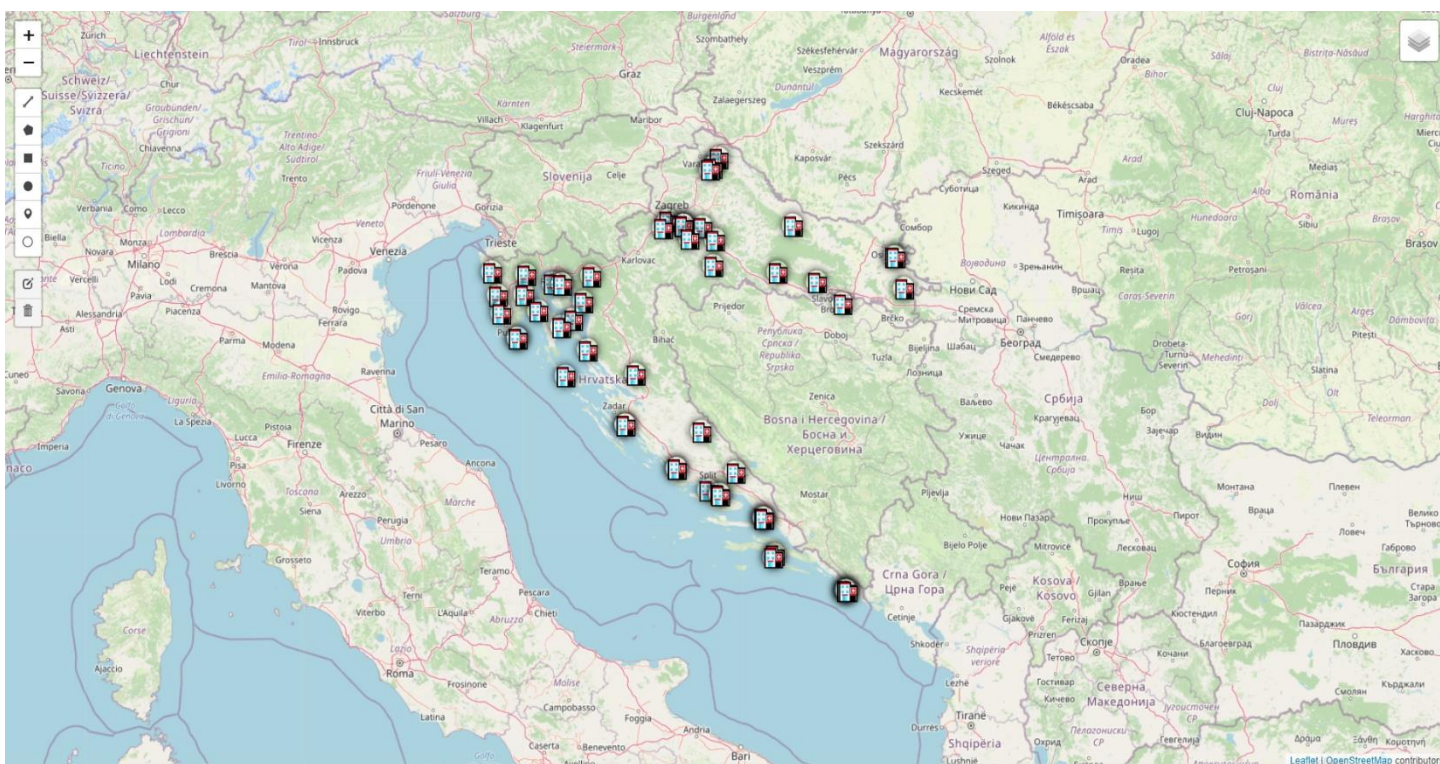
Osim navedenog, za prikaz podataka korištena je funkcija „*map()*“ iz JavaScripta koja je pozvana nad poljem „*icons*“. Kao drugi parametar korišten je „*id*“ po kojem je polje

mapirano. Kako se ne bi prikazivali prazni dijelovi teksta gdje nema podataka dodani su „if“ („?“) uvjeti.

```
{icons.map((icon, id) => (      josip, a month ago • initialized react-leaflet-draw
  <Marker
    position={[icon.lat, icon.lon]}
    icon={pinIcon}
    key={id}
    title={icon.header}
  >
    <Popup>
      <h1>📍 {icon.header}</h1>
      <h2>📍 Adresa: {icon.address}</h2>
      {icon.contact ? (
        <h3>
          📞 Kontakt: {icon.contact}
        </h3>
      ) : (
        <div></div>
      )}
      {icon.price ? <h4>💎 Cijena: {icon.price}</h4> : <div></div>}
      {icon.openHours ? (
        <h4>🕒 Radno vrijeme: {icon.openHours}</h4>
      ) : (
        <div></div>
      )}
      {icon.waitTime ? (
        <h4>🕒 Vrijeme čekanja: {icon.waitTime}</h4>
      ) : (
        <div></div>
      )}
      {icon.translate ? (
        <h4>📄 Prijevod(plaćanje): {icon.translate}</h4>
      ) : (
        <div></div>
      )}
      {icon.driveIn ? <h4>🚗 Drive in: {icon.driveIn}</h4> : <div></div>}
      <br />
    </Popup>
  </Marker>
```

Slika 45: „Marker“ i „Popup“ komponente u kojima su prikazani atributni podaci, programski kod aplikacije

Za prikaz ikona(*markera*) koji predstavljaju centre za testiranje, „*MarkerComponent*“ komponenta se poziva unutar „*MapComponent*“ komponente. Rezultat je sljedeći:



Slika 46: prikaz svih CoVid-19 centara za testiranje na razini Republike Hrvatske, primjer iz aplikacije

Izvor karte: <https://s.tile.openstreetmap.org/{z}/{x}/{y}.png>

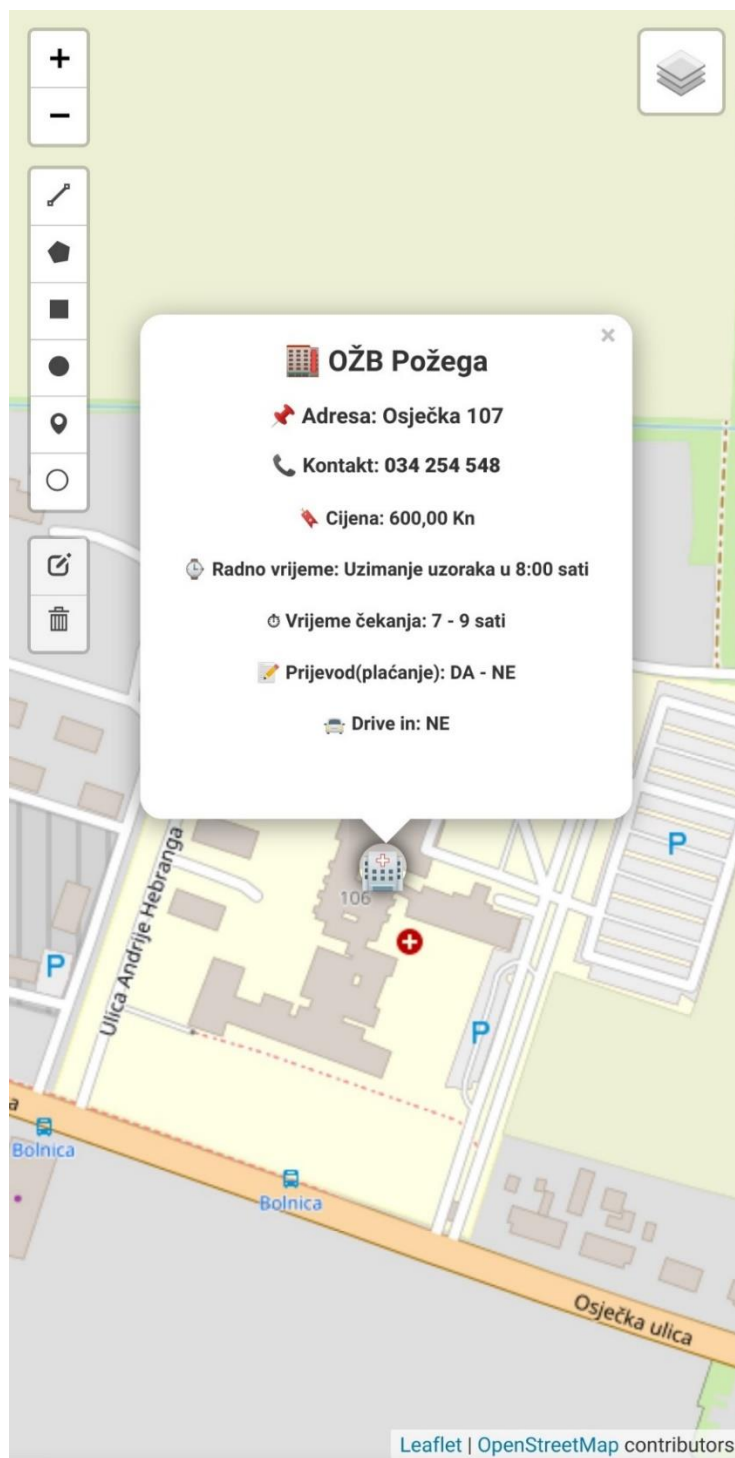
Svaki skočni prozor sa svojim vrijednostima posebno je stiliziran stilovima iz CSS datoteke. Nadalje, promijenjene su veličine fontova za uređaje sa rezolucijom manjom od 1080 piksela.

```
.leaflet-popup-content-wrapper h1 {
  font-size: 2.5vh;
}
.leaflet-popup-content-wrapper h2 {
  font-size: 2vh;
}
.leaflet-popup-content-wrapper h3 {
  font-size: 1.9vh;
}
.leaflet-popup-content-wrapper h4 {
  font-size: 1.8vh;
}
.leaflet-popup-content {
  min-width: 22rem;
  text-align: center;
}

@media only screen and (max-width: 1081px) {
  .leaflet-popup-content-wrapper h1 {
    font-size: 2vh;
  }
  .leaflet-popup-content-wrapper h2 {
    font-size: 1.5vh;
  }
  .leaflet-popup-content-wrapper h3 {
    font-size: 1.4vh;
  }
  .leaflet-popup-content-wrapper h4 {
    font-size: 1.3vh;
  }
  .leaflet-popup-content {
    min-width: 14rem;
  }
}
```

Slika 47: CSS stiliziranje skočnog prozora, programski kod aplikacije

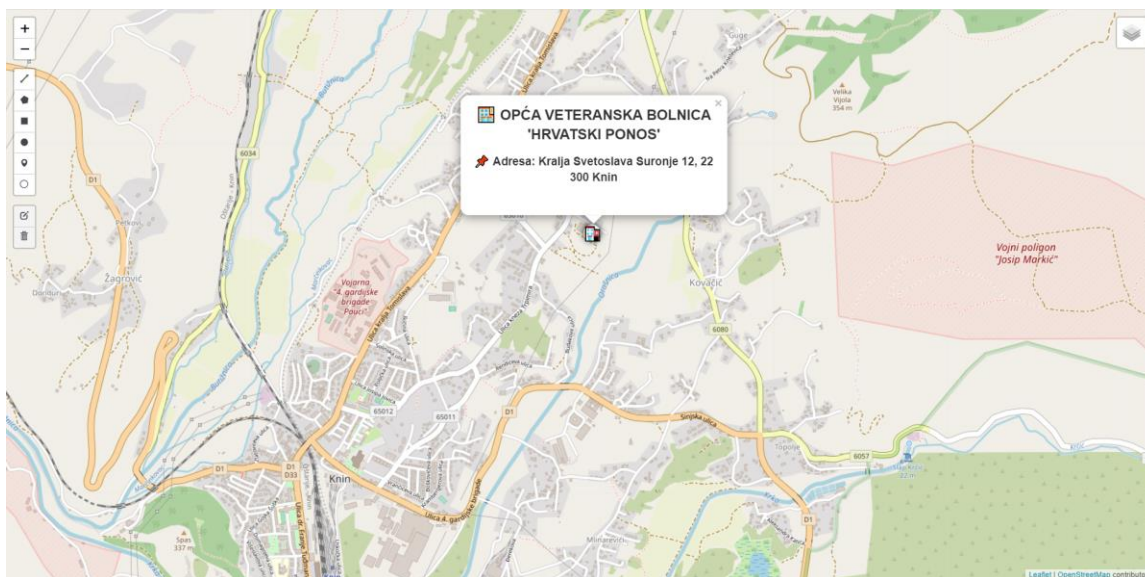
Izgled na mobilnim uređajima(preglednik Chrome):



Slika 48: OŽB Požega, izgled aplikacije na mobilnom uređaju

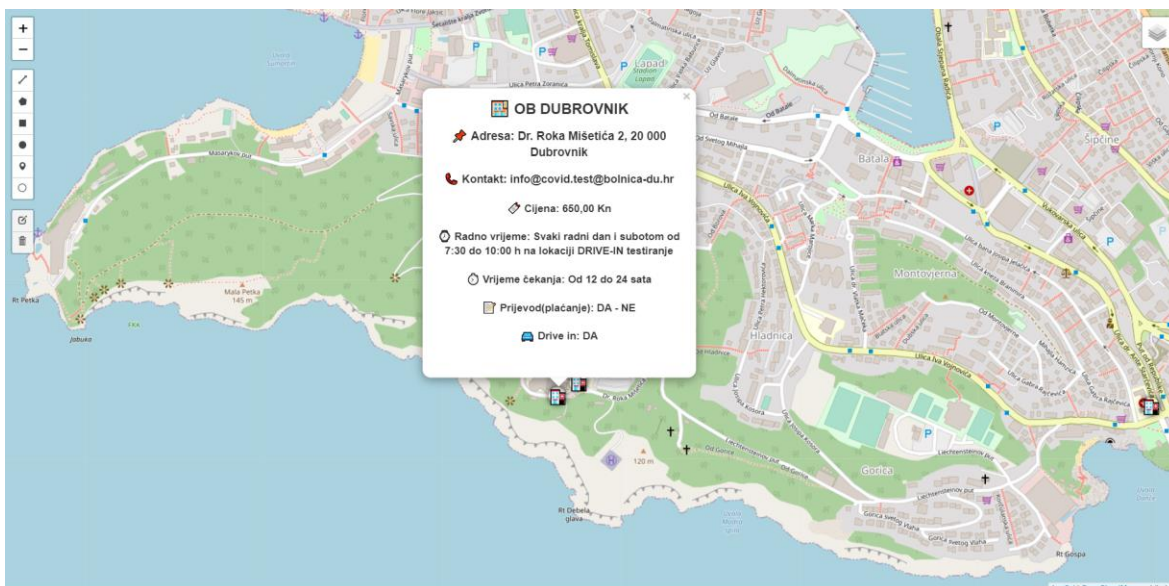
Izvor karte: <https://s.tile.openstreetmap.org/{z}/{x}/{y}.png>

Kao primjer, prikazan je skočni prozor centra za testiranje na u Kninu i centar za testiranje u Dubrovniku:



Slika 49: centar za testiranje Knin, primjer iz aplikacije

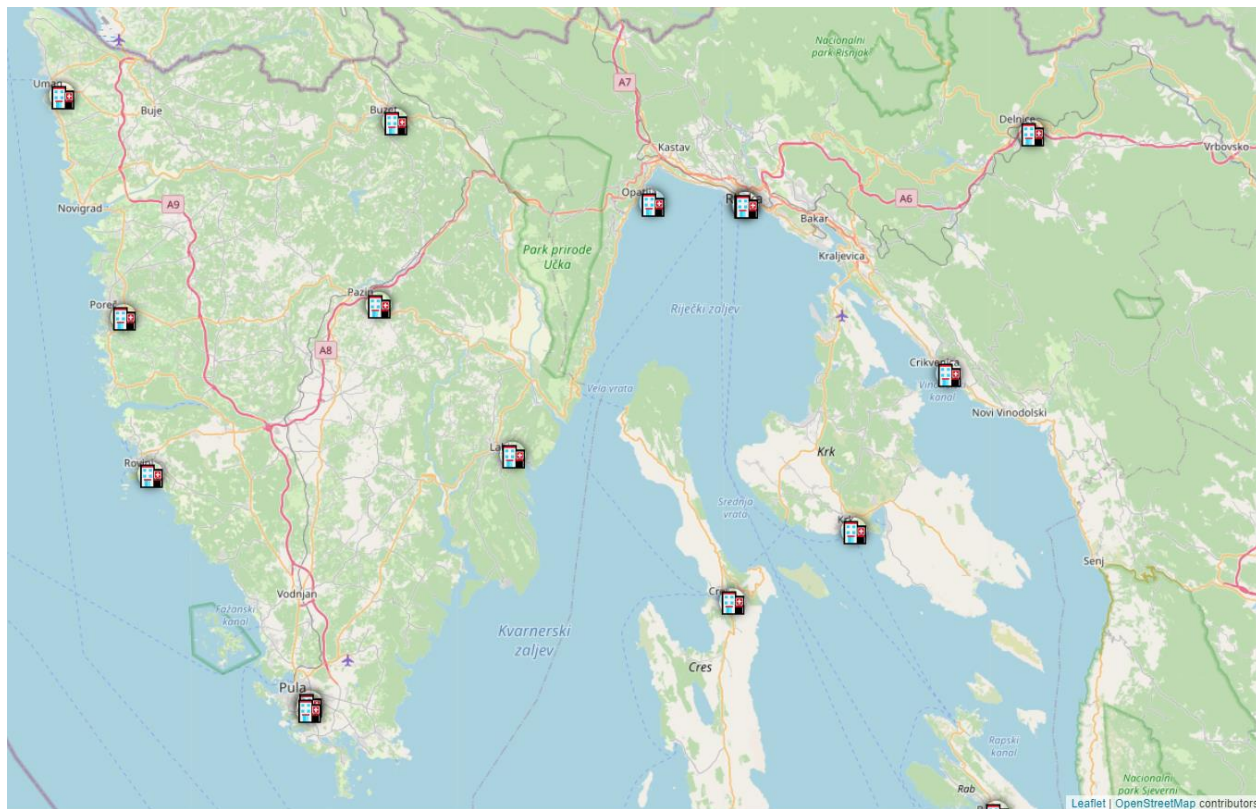
Izvor karte: <https://s.tile.openstreetmap.org/{z}/{x}/{y}.png>



Slika 50: centar za testiranje - OB Dubrovnik – primjer iz aplikacije

Izvor karte: <https://s.tile.openstreetmap.org/{z}/{x}/{y}.png>

Na karti je omogućeno kretanje i istraživanje, kao i već prije navedene značajke. Moguća je izrada brojnih drugih karti i prikaza. Kao primjer, prikazana je sljedeća karta zapadne Hrvatske:



Slika 51: centri za testiranje zapadnog dijela Hrvatske, primjer iz aplikacije

Izvor karte: <https://s.tile.openstreetmap.org/{z}/{x}/{y}.png>

Zaključak

U ovom radu korištene su tehnologije zasnovane na softveru otvorenog koda. Također, neke od tehnologija su poprilično robusne, te se mogu primijeniti za izradu većih sustava. U svakom slučaju izbor alata za rješavanje problema prikaza geoprostornih značajki u praksi se svodi na odluku kreatora. Razvijanje većeg geoinformacijskog sustava sa većim brojem podataka zahtijevao bi određene dorade. Ideja za razvijanjem ovakve aplikacije i ovoga rada javila se iz želje za sjedinjenjem elemenata geografsko informacijskih sustava i web tehnologija. Drugi dio aplikacijskog dijela rada služi rješavanju problema lakšeg pronalaženja centara za testiranje na *CoVid-19* i njihovih podataka: radno vrijeme, cijena, može li se test obaviti iz automobila i slično. Aplikacija bi zahtijevala manje vremena za razvoj ako bi postojali pripremljeni podaci, ili ako bi postojao API gdje bi se mogli ti podaci dohvatiti. Svrha ovoga rada bila je objasniti korištenje web tehnologija zajedno sa GIS-om i njegovim kartografskim elementima. Svrha drugog dijela rada bila je primijeniti web i GIS tehnologije opisane u prvom dijelu rada. Manje bitni dijelovi razvoja aplikacije poput povezivanja programskih datoteka nisu toliko detaljno objašnjeni, dok su svi ključni dijelovi programskog koda i način na koji funkcioniraju detaljnije objašnjeni. Radom se nastojalo predočiti kako za izradu raznih tipova karti i aplikacija zasnovanih na GIS elementima nije potrebna konvencionalna relacijska baza podataka. Mogućnosti za izradu ovakvih GIS aplikacija su neograničene, neovisno o tematici. Razvoj GIS aplikacija kroz softver otvorenog koda znatno je u porastu, što je pridonijelo njegovoj popularizaciji.

Literatura:

- [1] A Brief History of the Internet, University System of Georgia, https://www.usg.edu/galileo/skills/unit07/internet07_02.phtml, preuzeto: 04.09.2021.
- [2] What's difference between The Internet and The Web?, GeeksforGeeks, <https://www.geeksforgeeks.org/whats-difference-internet-web/> preuzeto: 04.09.2021.
- [3] Where the Web was born, CERN, <https://home.cern/science/computing/birth-web/short-history-web>, preuzeto: 04.09.2021.
- [4] Web Technology, GeeksforGeeks, <https://www.geeksforgeeks.org/whats-difference-internet-web/> preuzeto: 07.09.2021.
- [5] HTML: HyperText Markup Language, Mozilla, <https://developer.mozilla.org/en-US/docs/Web/HTML>, preuzeto: 23.08.2021.
- [6] Niederst Robbins, J.(2018.) *LEARNING WEB DESIGN: A BEGINNER'S GUIDE TO HTML, CSS, JAVASCRIPT, AND WEB GRAPHICS*. Peto izdanje. Sebastopol: O'Reilly Media
- [7] CSS selectors, Mozilla, https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Selectors, preuzeto: 23.08.2021.
- [8] JavaScript, Mozilla, https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Selectors, preuzeto: 23.08.2021.
- [9] TypeScript is JavaScript with syntax for types, Microsoft, <https://www.typescriptlang.org/>, preuzeto: 23.08.2021.
- [10] The Basics, Microsoft, <https://www.typescriptlang.org/docs/handbook/2/basic-types.html>, preuzeto: 23.08.2021.
- [11] Non-exception Failures, Microsoft, <https://www.typescriptlang.org/docs/handbook/2/basic-types.html>, preuzeto: 23.08.2021.
- [12] What Is React?, Facebook, <https://reactjs.org/tutorial/tutorial.html#what-is-react>, preuzeto: 23.08.2021.
- [13] Introducing JSON, ECMA, <https://www.json.org/json-en.html>, preuzeto: 24.08.2021.

- [14] Definitions, Internet Engineering Task Force (IETF),
<https://datatracker.ietf.org/doc/html/rfc7946>, preuzeto 24.08.2021.
- [15] Position, Internet Engineering Task Force (IETF),
<https://datatracker.ietf.org/doc/html/rfc7946>, preuzeto 24.08.2021.
- [16] Gregory, I. Ell, P.(2007), *Historical GIS: Technologies, Methodologies and Scholarship*,
 Cambridge: Cambridge university press
- [17] ATTRIBUTE DATA MODELS, PMR,
https://planet.uwc.ac.za/nisl/gis/web_page/page_20.htm, preuzeto 02.08.2021
- [18] Harvey, F.(2008), *A Primer of GIS: Fundamental Geographic and Carographic Concepts*,
 New York: The Guilford Press
- [19] RASTER DATA FORMATS, David J. Buckey,
https://planet.uwc.ac.za/nisl/BGIS/GIS_primer/page_17.htm, preuzeto: 05.08.2021.
- [20] VECTOR DATA FORMATS, David J. Buckey,
https://planet.uwc.ac.za/nisl/BGIS/GIS_primer/page_16.htm, preuzeto: 05.08.2021.
- [21] VECTOR AND RASTER - ADVANTAGES AND DISADVANTAGES, David J. Buckey,
https://planet.uwc.ac.za/nisl/BGIS/GIS_primer/page_19.htm, preuzeto: 05.08.2021.
- [22] Albreth, J.(2007), *Key Concepts and Techniques in GIS*, London: SAGE Publications Ltd
- [23] Hargitai, H.(2019), *Planetary Cartography and GIS*, Cham: Springer
- [24] Kumar, S. (2018), *Causes of Replacing Geographical Traditional Mapping into New World Digital Mapping through Geospatial Techniques*, DNPG College, Department of Geography
- [25] Representing spatial phenomena, Wikipedia, https://en.wikipedia.org/wiki/Map_symbol,
 preuzeto 20.08.2021.
- [26] Grafarend, E. Krumm, F.(2006) *Map projections. Cartographic information systems*,
 Stuttgart: Springer
- [27] Kennedy, M. Kopp, S.(2001), *Understanding Map Projections*, Redlands: ESRI

- [28] Conic Projections, Department of Geography and Environmental Science, Hunter College, <http://www.geo.hunter.cuny.edu/~jochen/gtech201/lectures/lec6concepts/Map%20coordinate%20systems/Conic%20projections.htm>, preuzeto 12.08.2021.
- [29] The Three Main Families of Map Projections, MathWorks, <https://www.mathworks.com/help/map/the-three-main-families-of-map-projections.html>, preuzeto 12.08.2021.
- [30] Topographic Maps, ICSM, <https://www.icsm.gov.au/education/fundamentals-mapping/types-maps/topographic-maps>, preuzeto 14.08.2021.
- [31] Thematic Maps, ICSM, <https://www.icsm.gov.au/education/fundamentals-mapping/types-maps/thematic-maps>, preuzeto 14.08.2021.
- [32] Scharl, A. Tochtermann, K.(2007) , *The Geospatial Web: How Geobrowsers, Social Software and the Web 2.0 are Shaping the Network Society*, Graz: Springer
- [33] Introduction to WMS, OGC, <https://www.ogc.org/standards/wms/introduction>, preuzeto 22.08.2021
- [34] Why tiled maps?, The Pennsylvania State University, <https://www.e-education.psu.edu/geog585/node/706>, preuzeto 18.08.2021.
- [35] Map and Tile Coordinates, Google, <https://developers.google.com/maps/documentation/javascript/coordinates>, preuzeto 19.08.2021.
- [36] Tambassi, T.(2019) *The Philosophy of GIS*, Parma: Springer
- [37] Google Maps Platform overview, Google, <https://cloud.google.com/maps-platform/>, preuzeto 01.08.2021.
- [38] Maps and location for developers, MapBox, <https://www.mapbox.com/>, preuzeto 05.08.2021.
- [39] OpenStreetMap, OpenStreetMap, <https://en.wikipedia.org/wiki/OpenStreetMap>, preuzeto 02.08.2021.
- [40] Overview, Leaflet, <https://leafletjs.com/index.html>, preuzeto 03.08.2021.
- [41] ArcGIS API for JavaScript, <https://developers.arcgis.com/javascript/3/jshelp/>, preuzeto 04.08.2021.

- [42] Overview, OpenLayers, <https://openlayers.org/>, preuzeto 04.08.2021.
- [43] Google Maps Platform Documentation, Google, <https://developers.google.com/maps/documentation>, preuzeto 04.08.2021.
- [44] Overview, Google, <https://developers.google.com/maps/documentation/javascript/overview>, preuzeto 04.08.2021.
- [45] EPSG:3857, OpenStreetMap, <https://wiki.openstreetmap.org/wiki/EPSC:3857>, preuzeto 09.08.2021.
- [46] Core concepts, React-Leaflet, <https://react-leaflet.js.org/docs/start-introduction/>, preuzeto 10.08.2021.
- [47] OpenTopoMap, OpenStreetMap, <https://opentopomap.org/about>, preuzeto 14.08.2021.
- [48] Humanitarian map style, OpenStreetMap, https://wiki.openstreetmap.org/wiki/Humanitarian_map_style, preuzeto 14.08.2021.
- [49] wmflabs Hillshading, OpenStreetMap, https://wiki.openstreetmap.org/wiki/Tile_servers, preuzeto 14.08.2021.
- [50] wmflabs OSM B&W, OpenStreetMap, https://wiki.openstreetmap.org/wiki/Tile_servers, preuzeto 14.08.2021.
- [51] Hike & Bike Map, OpenStreetMap, [https://wiki.openstreetmap.org/wiki/Hike %26 Bike Map](https://wiki.openstreetmap.org/wiki/Hike_%26_Bike_Map), preuzeto 14.08.2021.
- [52] wmflabs OSM no labels, OpenStreetMap, https://wiki.openstreetmap.org/wiki/Tile_servers, preuzeto 14.08.2021.
- [53] ÖPNVKarte, MeMoMaps, <https://wiki.openstreetmap.org/wiki/%C3%96PNVKarte>, preuzeto 14.08.2021.
- [54] Toner, Stamen, <http://maps.stamen.com/#toner/12/37.7706/-122.3782>, preuzeto 14.08.2021.
- [55] EditControl API, React-Leaflet-Draw, <https://www.npmjs.com/package/react-leaflet-draw/v/0.19.5>, preuzeto 09.08.2021.
- [56] Leaflet Draw API reference, Leaflet and Leaflet Organization, <https://leaflet.github.io/Leaflet.draw/docs/leaflet-draw-latest.html>, preuzeto 09.08.2021.

Popis slika:

Slika 1: primjer HTML tag-a

Slika 2: primjer jednostavne HTML stranice

Slika 3: struktura CSS pravila

Slika 4: primjer CSS pravila za stiliziranje za tag <h1>

Slika 5: poziv metode „location“ nad objektom „user“ u JavaScriptu

Slika 6: poziv metode „location“ nad objektom „user“ u JavaScriptu

Slika 7: jednostavni JSON objekt, sa poljem „cars“

Slika 8: primjer GeoJSON objekta za jednu točku

Slika 9: primjer rasterskog i vektorskog modela prikaza

Slika 10: proces prikaza prostornih podataka

Slika 11: Merkatorova projekcija

Slika 12: Sinusoidalna projekcija

Slika 13: ekvidistantna cilindrična projekcija

Slika 14: gnomonska projekcija zemlje sa centrom na sjevernom polu.

Slika 15: Craig-ova retroazimutalna projekcija sa centrom u Meki

Slika 16: Robinsonova projekcija, primjer kompromisne projekcije

Slika 17: cilindrična projekcija

Slika 18: primjer pseudocilindrične projekcije – Robinsonova projekcija

Slika 19: konusna projekcija

Slika 20: Bonne-ova projekcija, primjer pseudokonusne projekcije

Slika 21: Azimutna projekcija. Centar – Sjeverni pol

Slika 22: topografska karta Pule, primjer iz aplikacije

Slika 23: prikaz tematske(transport) karte Pule, primjer iz aplikacije

Slika 24: „MapContainer“ komponenta, primjer iz aplikacije

Slika 25: „LayersControl“ i „TileLayer“ komponente za dohvat i prikaz karata, programski kod aplikacije

Slika 26: „LayersControl“ ikona za promjenu karata

Slika 27: izbornik za mijenjanje karata, primjer iz aplikacije

Slika 28: OpenStreetMap classic, grad Pula i okolica, primjer iz aplikacije

Slika 29: OpenTopoMap, grad Pula i okolica, primjer iz aplikacije

Slika 30: Humanitarian Map Style, grad Pula i okolica, primjer iz aplikacije

Slika 31: wmf labs Hillshading map, Istra, Primorje i Kvarner, primjer iz aplikacije

Slika 32: OSM Black and White, grad Pula i okolica, primjer iz aplikacije

Slika 33: wmf labs Hike and Bike, grad Vodnjan, primjer iz aplikacije

Slika 34: wmf labs OSM no labels, grad Pula, primjer iz aplikacije

Slika 35: Öpnvkarte Transport map, grad Osijek, primjer iz aplikacije

Slika 36: Stamen Toner map, grad Venecija, Italija, primjer iz aplikacije

Slika 37: „React Leaflet Draw“ izbornik, primjer iz aplikacije

Slika 38: „EditControl“ (React Leaflet Draw) komponenta, programski kod aplikacije

Slika 39: iscrtani vektorski elementi, „React Leaflet draw“, primjer iz aplikacije

Slika 40: objekt sa podacima o centru za testiranje KBC Osijek tipa JSON, programski kod aplikacije

Slika 41: „MarkerComponents“ interface, definicija tipova, programski kod aplikacije

Slika 42: primjeri JSON polja objekata „MarkerComponents“, programski kod aplikacije

Slika 43: Objekt „pinIcon“, definicije markera.

Slika 44: CSS stilovi za ikone, programski kod aplikacije

Slika 45: „Marker“ i „Popup“ komponente u kojima su prikazani atributni podaci, programski kod aplikacije

Slika 46: prikaz svih *CoVid-19* centara za testiranje na razini Republike Hrvatske, primjer iz aplikacije

Slika 47: CSS stiliziranje skočnog prozora, programski kod aplikacije

Slika 48: OŽB Požega, izgled aplikacije na mobilnom uređaju

Slika 49: centar za testiranje Knin, primjer iz aplikacije

Slika 50: centar za testiranje - OB Dubrovnik – primjer iz aplikacije

Slika 51: centri za testiranje zapadnog dijela Hrvatske, primjer iz aplikacije

Popis tablica:

Tablica 1: Podaci o centrima za testiranje na *CoVid*

Sažetak

Cilj ovog diplomskog rada bio je izraditi GIS aplikaciju uz pomoć modernih web tehnologija. Osim pojmova GIS i WebGIS, pojašnjeni su i osnovni pojmovi kartografije i web kartiranja. Aplikacija se sastoji od dva dijela, a izrađena je web tehnologijama: HTML, CSS, JavaScript/TypeScript, React, React-Leaflet. Prvi dio aplikacije jesu razne web karte i crtanje elemenata vektorske grafike na njima, dok drugi dio aplikacije jest geovizualizacija lokacija za testiranje na virus CoVid-19.

Ključne riječi: web tehnologije, GIS, kartografija, geovizualizacija

Summary

In this thesis, the goal is to create a GIS application with the help of modern web technologies. In addition to GIS and WebGIS, cartography and web mapping terms are explained. The application consists of two parts, and it is created with the use of these web technologies: HTML, CSS, JavaScript/TypeScript, React, React-Leaflet. The first part of the application are web maps and tools for drawing vector graphic elements on top of them, while the second part of the application provides geovisualization of locations for testing on the CoVid-19 virus.

Key words: web technologies, GIS, cartography, geovisualization