Izrada aplikacije za grafički prikaz podataka prikupljenih uporabom senzorske mreže

Šapina, Tomislav

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET

Sveučilišni studij

Izrada aplikacije za grafički prikaz podataka prikupljenih uporabom senzorske mreže

Diplomski rad

Tomislav Šapina

Osijek, 2016

Sadržaj

1. UVOD 1
2. KORIŠTENE TEHNOLOGIJE 2
2.1. Front-end razvoj 2
2.1.1. HTML 2
2.1.2. CSS 4
2.1.3. JavaScript 6
2.1.4. AJAX 6
2.2. Back-end razvoj 7
2.2.1. PHP 8
2.2.2. MySQL 9
3. PROGRAMSKI OKVIRI I ALATI 10
3.1. Bootstrap 10
3.2. jQuery 11
3.3. Date range picker 11
3.4. JSON 12
3.5. GeoJSON 13
3.6. Leaflet 16
3.7. Dygraphs 18
3.8. XAMPP 19
3.9. PhpMyAdmin 20
3.10. Sublime Text 21
4. IMPLEMENTACIJA I PRIKAZ APLIKACIJE 22
4.1. Rad sa bazom podataka 22
4.2. Opis datoteka aplikacija 25
4.3. Prikaz podataka na karti Hrvatske 28
4.3.1. Opis realizacije 31
4.4. Prikaz podataka na linijskom grafu 38
5. ZAKLJUČAK 42
LITERATURA 43
SAŽETAK 44
ŽIVOTOPIS 45
PRILOG A. Programski kod aplikacije 46

1.UVOD

U ovom radu prikazana je izrada aplikacije za grafički prikaz podataka prikupljenih uporabom senzorske mreže. Podaci koji se prikazuju u aplikaciji predstavljaju parametre koji se prikupljaju putem meteo stanica. Parametri koji su prikupljani su vlažnost tla (mjerenja sa šest različitih dubina), temperatura zraka, vlažnost zraka, brzina vjetra i količina padalina. U svrhu izrade ovog rada podaci će se generirati prema očekivanim mogućim vrijednostima za odabrane parametre. Podaci su generirani za četiri meteo stanice koje će se postaviti na proizvoljno odabrane geografske lokacije u Republici Hrvatskoj. Korisniku je omogućen prikaz podataka na karti Hrvatske te na linijskom grafu.

Aplikacija je izrađena uporabom suvremenih web tehnologija. Korišten je HTML5 (*HyperText Markup Language*) kao osnova i jezik za prikaz elemenata na ekranu. Uz HTML5 uvijek je vezan CSS3 (*Cascading Style Sheets*) jezik koji opisuje način prikaza HTML elemenata u web pregledniku. Uz HTML5 i CSS3 na klijentskoj se strani još koristi i skriptni programski jezik JavaScript. Za rad na poslužiteljskoj strani odabran je PHP (*PHP: Hypertext Preprocessor*). Za rad sa bazom podataka korišten je SQL (*Structured Query Language*). Za pristup poslužiteljskoj strani korišten je asinkroni pristup što omogućuje AJAX (*Asynchronous JavaScript and XML*) tehnologija. Za lakšu i efektivniju izradu korišteni su razni JavaScript i CSS programski okviri. Aplikacija je smještena na web poslužitelju s instaliranom Apache web poslužiteljskom aplikacijom.

Glavni dio rada je podijeljen na tri dijela. U drugom poglavlju naziva „*Korištene tehnologije*“ opisane su teorijske podloge korištenih tehnologija. U trećem poglavlju „*Programski okviri i alati*“ opisani su programski okviri i alati koji se koriste pri implementaciji aplikacije. Četvrto poglavlje nosi naziv „*Implementacija i prikaz aplikacije*“ gdje je detaljno prikazan proces izrade aplikacije te rezultati izrade aplikacije. Zaključna razmatranja dana su u petom poglavlju.

2.KORIŠTENE TEHNOLOGIJE

2.1. Front-end razvoj

Razvoj aplikacija sastoji se od dvije osnovne podjele. Razvoj na prednjoj strani (eng. *front-end*) i razvoj u pozadini (eng. *back-end*). U nastavku rada za razvoj na prednjoj strani bit će korišten termin front-end te će za razvoj u pozadini biti korišten termin back-end. Kroz rast programerske industrije proteklih godina stvara se sve veća razlika između razvoja na front-end strani i razvoja na back-end strani. Danas se razvijatelji (eng. *developers*) aplikacija ovisno o tehnologijama koje koriste za razvoj, dijele na frontend razvijatelje i na backend razvijatelje što nije bio slučaj prije deset godina.

Front-end dio koda se izvodi na klijentskoj strani, preciznije na web pregledniku ako je riječ o web aplikacijama. To je dio koji korisnici mogu vidjeti i stvoriti interakciju s istim. Vizualni dio frontenda uključuje dugmad, forme, tipografiju, animacije i sadržaj. U osnovi to je sve što korisnik može vidjeti kada koristi aplikaciju. Logički dio front-end razvoja služi za dodavanje dinamičnosti aplikacije te komunikacije s back-end dijelom. U kontekstu web aplikacija cilj front-end razvoja je izrada brzih stranica koje su razumljive i ugodne oku korisnika. Tehnologije koje spadaju u front-end razvoj su HTML, CSS, Javascript i AJAX koje će biti opisane u nastavku rada. Osim navedenih tehnologija valja napomenuti da i sama izrada dizajna spada pod front-end dio.

2.1.1. HTML

HTML dolazi od engleskih riječi HyperText Markup Language i najrasprostanjeniji je jezik za izradu web stranica. Autor prve verzije ovog jezika je Berners-Lee koja je neslužbeno objavljena 1991. godine. U to vrijeme jezik je bio dosta ograničen. Današnja verzija koja će biti korištena u ovom radu je HTML 5 koja je izašla 2014. godine od strane W3C (*World Wide Web Consortium*). [1]

U tablici 2.1. prikazan je povijesni pregled HTML verzija.

Tablica 2.1. Povijesni pregled razvoja HTML jezika

[Izvor: http://www.w3schools.com/html/html_intro.asp]

Verzija	Godina
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.0.1	1999
XHTML	2000
HTML5	2014

HTML dokumenti sadrže ekstenziju „html“ ili „htm“. Temeljni građevni element svakog HTML dokumenta su oznake (eng. *tags*) kojima se opisuje HTML dokument. Unutar svake oznake se nalazi naziv HTML elementa. [1]

Kod 2.1. Primjer HTML oznake

```
<p>ovo je paragraf</p>
```

Navedena oznaka u kodu 2.1. opisuje odlomak teksta. Početna i završna oznaka sadrže naziv oznake, a završna oznaka mora imati kosu crtu prije naziva. U sljedećem primjeru koda 2.2. naveden je HTML kod za jednostavnu web stranicu.

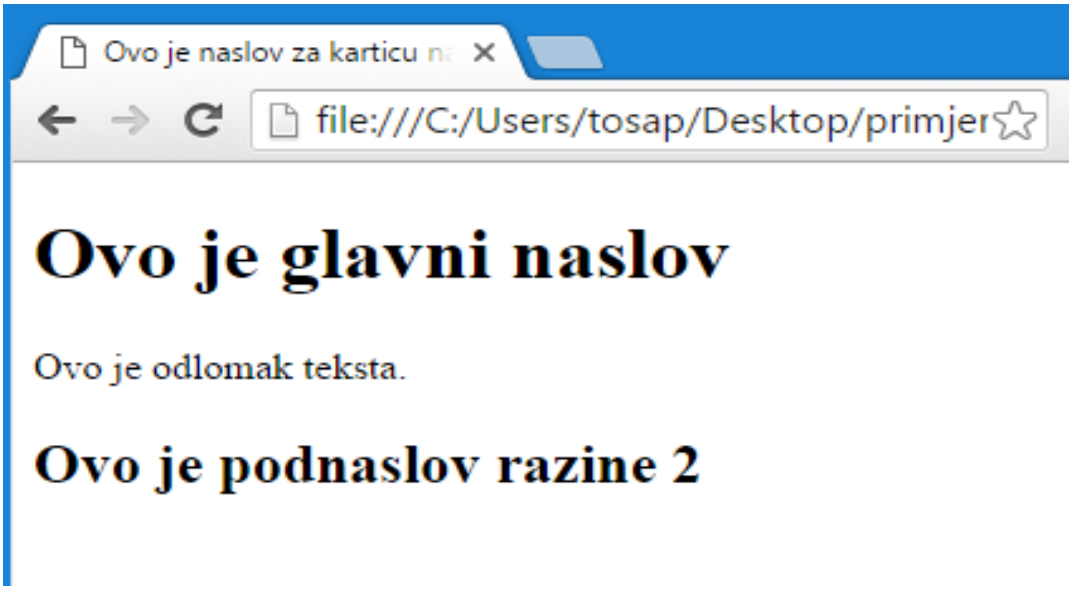
Kod 2.2. Primjer jednostavne HTML stranice

```
<!doctype html>
<html>
<head>
<title>Ovo je naslov za karticu na web pregledniku</title>
</head>
<body>
<h1>Ovo je glavni naslov</h1>
<p>Ovo je odlomak teksta.</p>
<h2>Ovo je podnaslov razine 2</h2>
```

```
</body>
</html>
```

Svaka web stranica uobičajeno počinje i završava sa <*html*> oznakom. Prije početne HTML oznake navedena je deklaracija <*!DOCTYPE*> koja govori pregledniku u kojoj verziji HTML-a je pisana web stranica. Unutar <*html*> oznake nalaze se dvije cjeline head i body koje su definirane <*head*> i <*body*> oznakama. Head dio sadrži informacije o naslovu dokumenta koji se prikazuje na kartici preglednika, o stilovima dokumenta , o skriptama dokumenta i o drugim meta informacijama. Unutar body dijela nalaze se sve oznake koje opisuju sav sadržaj HTML dokumenta kao što su tekst, linkovi, slike , tablice itd. [1]

U nastavku je slika 2.1. koja je rezultat koda 2.2.



Slika 2.1. Prikaz jednostavnog HTML dokumenta [Izvor: autor]

2.1.2. CSS

CSS je skraćenica za Cascading Style Sheets. CSS je stilski jezik kojim se definira prezentacija HTML dokumenta. Za CSS jezik je zaslužan W3C, a izumljen je iz razloga da bi unaprijedio izradu web stranica. Pojedinim HTML elementima se pomoću CSS jezika dodjeljuju stilovi te se tako definira grafička prezentacija web dokumenta. Pomoću CSS jezika moguće je definirati fontove, boje, margine, okvire, veličine i ostala svojstva. [2]

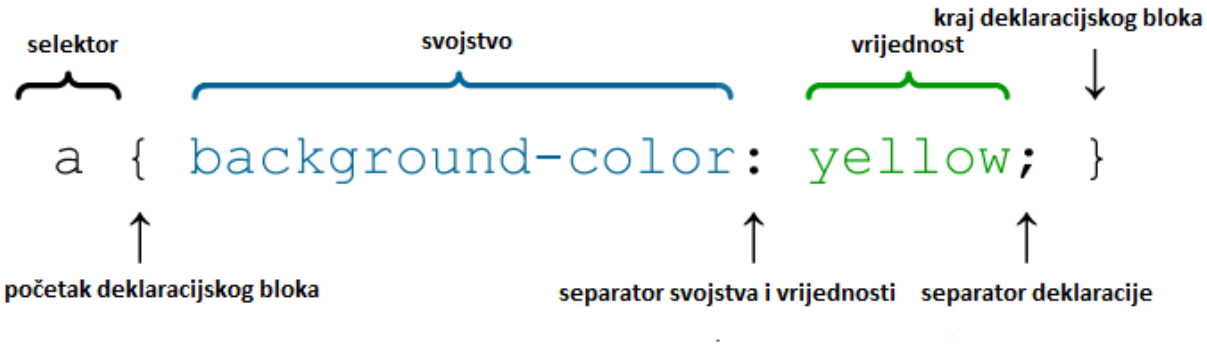
U tablici 2.2. prikazan je povijesni pregled CSS verzija.

Tablica 2.2. Povijesni pregled razvoja CSS jezika

[Izvor: http://www.w3schools.com/css/css_intro.asp]

Verzija	Godina
CSS1	1996
CSS2	1998
CSS 2.1	2004
CSS 3	2012

CSS sintaksa sastoji se od selektora i deklaracijskog bloka. Selektor pokazuje na HTML element ili skupinu HTML elemenata nad kojima će se primjeniti svojstva opisana u deklaracijskom bloku. Deklaracijski blok može imati jedno ili više svojstva, a uz svako svojstvo je definirana vrijednost svojstva. [2]



Slika 2.2. Primjer CSS sintakse [Izvor: autor]

Na slici 2.2. je prikazan jednostavni primjer CSS sintakse. CSS kod se najčešće piše u odvojenu datoteku ekstenzije „css“ čija se putanja definira u head dijelu HTML dokumenta. Moguć je i način pisanja CSS koda unutar samog HTML dokumenta između oznaka <*style*>. Treća mogućnost pisanja CSS-a je unutar atributa style željenog HTML elementa.

2.1.3. JavaScript

JavaScript je programski jezik koji se izvršava u web pregledniku. Podržan je od strane svih web preglednika. Izumio ga je Brendan Eich iz tvrtke Netscape 1996. godine. 1997. nastaje prva inačica standarda za taj jezik razvijena od strane ECMA (*European Computer Manufacturers Association*). Cilj kreiranja JavaScript jezika bio je dodati interaktivnost HTML stranicama. JavaScript ima sličnu sintaksu kao C++, C, C#, Java itd. Neke od mogućnosti koje nam JavaScript nudi su programiranje u okviru HTML stranica, pretvaranje dinamičkog teksta u HTML stranicu, reagiranje na događaje, čitanje i pisanje HTML elementa, validiranje (provjeru ispravnosti i vjerodostojnosti) podataka, detektiranja preglednika kojeg korisnik upotrebljava, kreiranje kolačića itd [3].

JavaScript kod se može pisati unutar HTML dokumenta pod oznakom *<script type="text/javascript">* . Drugi, češće korišteni način pisanja JavaScript koda je pisanje u odvojenu datoteku ekstenzije „.js“ koja se povezuje sa HTML dokumentom unutar oznake *<head>*. Danas se često uz JavaScript vežu pojmovi kao što su jQuery, angluar js, react js. To su programski okviri pisani u JavaScriptu koji su nastali kako bi olakšali korištenje JavaScripta te unaprijedili front-end razvoj. Osim na klijentskoj strani,u novije vrijeme javacript se sve više koristi i na poslužiteljskoj strani.

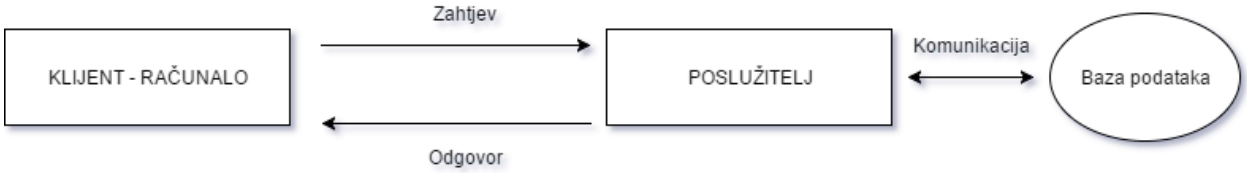
2.1.4. AJAX

AJAX je skraćenica od engleske riječi Asynchronous JavaScript and XML (*EXtensible Markup Language*). To je tehnika koja omogućava izradu bržih i dinamičnijih web stranica. Omogućava dohvaćanje podataka s poslužitelja klijentu bez osvježavanja web stranice, što je ogroman korak za web tehnlogije. AJAX nije tehnologija zasebno nego je grupa tehnologija. Temelji se na internet standardima, a koristi *XMLHttpRequest* objekt za dohvaćanje podataka s poslužitelja i Javascript za prikaz ili korištenje podataka. [4]

2.2. Back-end razvoj

Back-end web aplikacija obično se sastoji od tri djela: poslužitelj, aplikacija i baza podataka. Back-end dio koda odgovoran je za stvari kao što su logika aplikacije, komunikacija s bazom i izvedba aplikacije. Generalno gledano, većina koda neophodna za rad aplikacije je smještena na poslužiteljskoj strani. Back-end kod se pokreće na poslužitelju, što znači da back-end razvijatelji osim razumijevanja programiranja i baze podataka moraju poznavati i arhitekturu poslužitelja. Programski jezici za rad na poslužitelju su PHP, Ruby, Java, Python, .net itd. Za rad s podacima najčešće se koriste sustavi kao MySQL, Oracle i SQL. HTTP (*Hyper Text Transfer Protocol*) je protokol koji omogućuje komunikaciju između klijenta i poslužitelja te je najčešća metoda za prijenos informacija s poslužitelja na klijenta i obrnuto.

Kao i kod front-end razvoja tako i kod back-end razvoja postoje mnogi programski okviri koji olakšavaju i ubrzavaju razvoj dinamičkih web aplikacija. Još valja napomenuti da tom dijelu koda korisnik aplikacije nema pristup te ga ne može mijenjati. Za aplikaciju prikazanu u radu koristit će se skriptni jezik PHP i MySQL baza podataka.



Slika 2.3. Komunikacija klijent – poslužitelj [Izvor: autor]

2.2.1. PHP

Naziv PHP je skraćenica od engleskog PHP:Hypertext preprocessor. PHP je široko rasprostranjeni programski jezik na strani poslužitelja dizajniran u svrhu izrade dinamčnih web stranica. Razvijen je u C programskom jeziku pa mu je sintaksa slična onoj iz programskog jezika C. PHP radi na različitim operacijskim sustavima. Prvu verziju PHP je razvijo Rasmus Lerdorf 1994. Trenutno je aktivna verzija 7. Tek u 4. verziji PHP poprima karakteristike objektno orijentiranog jezika. [5]

U tablici 2.3. prikazan je povijesni pregled PHP verzija.

Tablica 2.3. Povijesni pregled razvoja PHP jezika [Izvor: https://secure.php.net/releases/]

PHP 1 ,2	1995 -1996
PHP 3	1997
PHP 4	2000
PHP 5	2004
PHP 6	2009
PHP 7	2014

Popularan je jer podržava mnoge baze podataka kao MySQL, Oracle, PostfreSQL, Sybase its. PHP je besplatan (eng. *open source*) te ga koristi oko 80% web stranica. PHP stranice su općenito HTML stranice s PHP naredbama ugrađenim u njih. Poslužitelj obrađuje PHP naredbe te vraća rezultat obrade na klijent stranu. Rezultat obrade je HTML dokument koji se prikazuje korisniku u pregledniku. Kako bi se PHP kod držao odvojen od HTML koda potrebno ga je umetnuti unutar specijalnih tagova *<?php* i *?>*. U nastavku pod kod 2.3. naveden najosnovniji primjer PHP koda koji ispisuje neki tekst.

Kod 2.3. Primjer jednostavnog PHP koda

```
<?php
echo "Ovo je najosnovniji primjer PHP koda!";
?>
```

Trenutno je dostupno mnoštvo PHP programskih okvira koji sadržavaju najčešće korištene funkcionalnosti kao što je validacija formi, rad s bazom podataka, upravljanje sesijama, upravljanje kolačićima, rad sa e- poštom, kalendar itd. Većina radnih okvira se bazira na MVC (eng. *Model View Controller*) arhitekturi. MVC arhitektura omogućava rad u timu te je potrebno pisati manje koda, što u konačnici ubrazava razvoj aplikacije. Neki od popularnih programskih okvira su Zend Framework, Cake PHP, Symfony, Laravel, Code Igniter itd.

U aplikaciji opisanoj u radu nije korišten PHP programski okvir jer uglavnom nisu potrebne gore navedene funkcionalnosti. U aplikaciji opisanoj u radu je korišten PHP za spajanje na MySQL bazu podataka i dohvaćanje podataka mjerenja. MySQL baza opisana je u sljedećem potpoglavlju.

2.2.2. MySQL

Baza podataka je skup međusobno povezanih podataka koji su na raspolaganju korisnicima za pregledavanje, pretraživanje, brisanje, nadopunjavanje i ispravljanje. Za izradu ove aplikacije korišten je besplatan sustav za upravljanje bazom podataka MySQL. MySQL baza je relacijskog tipa i moguće ju je koristiti u kombinaciji sa PHP-om kao i drugim brojnim programskim jezicima. Najčešće je izbor za aplikacije i web stranice otvorenog koda te se najčešće distribuira u poslužiteljskim inačicama Linux distribucija, iako postoje i verzije za Windowse i MacOS. SQL (*Structured Query Language*) je standardni jezik za pristup i upravljanje bazama podataka. Neke od osnovnih funkcija SQL jezika su dohvaćanje podataka iz baze, umetanje novih podataka, brisanje podataka, izmjena podataka, kreiranje novih tablica itd. U aplikaciji je korišten SQL u kombinaciji s PHP funkcijama.

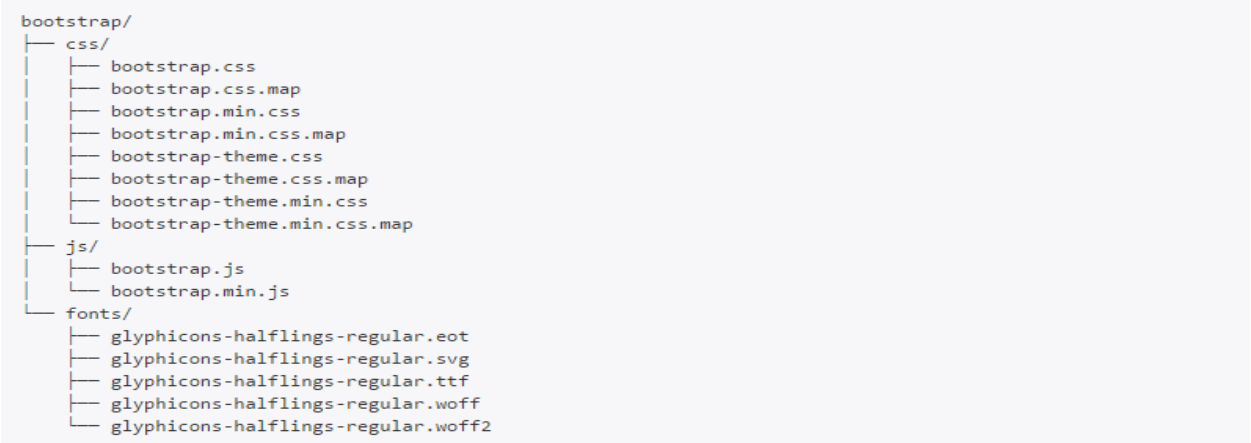
3.PROGRAMSKI OKVIRI I ALATI

Programski okviri su dizajnirani kako bi ubrzali i pojednostavili izradu aplikacija. Omogućuju one funkcionalnosti koje se često koriste te omogućuju izmjenu istih. Kada govorimo o front-end razvoju, najčešći programski okvir je Bootstrap koji sadrži vlastite i gotove CSS i JavaScript datoteke koje frontend razvijatelji koriste pri izradi web stranica. Na taj način se smanjuje vrijeme potrebno za razvoj jedne web stranice jer nije neophodno pisati vlastiti CSS i JavaScript ispočetka, već se postojeći kod može koristiti i mijenjati po potrebi. Najčešće korišteni programski okviri u back-end razvoju navedeni su u potpoglavlju 2.2.1 gdje se opisuje PHP.

3.1. Bootstrap

Bootstrap je najpopularniji HTML, CSS i JavaScript programski okvir za razvoj responzivnih web stranica i web aplikacija razvijen od strane Marka Otta i Jacoba Thorntona. Čini front-end razvoj bržim i lakšim. Sadrži HTML i CSS predloške za tipografiju, navigaciju, dugmad, liste, forme i druge komponente korisničkog sučelja. Također, sadrži i nekoliko JavaScript komponenata u obliku jQuery dodataka koje omogućuju dodatne elemente korisničkog sučelja kao što su dijaloški okviri, tooltips i slideri. Isto tako, proširuju funkcionalnosti postojećih elemenata korisničkog sučelja kao što je automatsko popunjavanje polja za unos. [6]

Bootstrap je dobro dokumentiran što omogućava bržu i lakšu implementaciju. Razvijatelji mogu nadograditi i prilagoditi Bootstrap svojim potrebama tako što mogu odabrati komponente koje žele koristiti u svom projektu. Takva vrsta odabira uvelike olakšava i proširuje primjenu ovog programskog okvira. Na slici 3.1 prikazana je struktura Bootstrap datoteka.



Slika 3.1. Prikaz strukture Bootstrap datoteka [Izvor: autor]

3.2. jQuery

JQuery je JavaScript biblioteka dizajnirana kako bi se olakšalo front-end pisanje koda. Koristi ju više od 60% od deset milijuna najposjećenijih web stranica. Kreirana je od strane Johna Resiga 2006. godine. Omogućava i pojednostavljuje rad s DOM (*Document Object Model*) elementima, kreiranje animacija, rukovanje događajima itd. Također, olakšava uporabu AJAX tehnologije. [7]

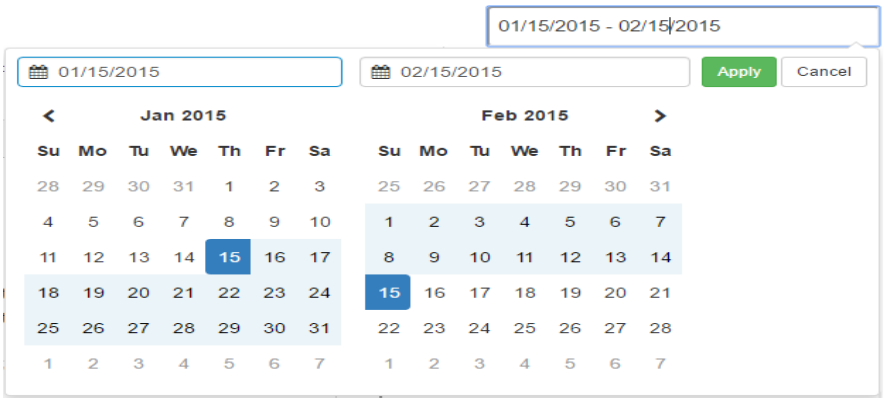
Uz pomoć ove biblioteke je nastao ogroman broj dodataka (eng. *plugins*), besplatnih i komercijalnih. Najpopularniji internet preglednici podržavaju jQuery-a. Da bi koristili jQuery funkcije potrebno je uključiti jQuery file u head dio HTML dokumenta.

3.3. Date range picker

Odabir raspona podataka (eng. *Date range picker*) je JavaScript komponenta koja omogućuje odabir ranga datuma. Radi uz pomoć jQuery biblioteke i Bootstrap programskog okvira. [8]

Kod 3.1. Primjer postavljanja DateRange pickera

```
<input type="text" name="daterange" value="01/01/2015 - 01/31/2015" />
<script type="text/javascript">
$(function() {
    $('input[name="daterange"]').daterangepicker();
});
</script>
```



Slika 3.2. Rezultat koda pod 3.1. u web pregledniku [Izvor: autor]

Navedeni kod 3.1. predstavlja primjer sa službene stranice te je navedena slika 3.2. koja predstavlja rezultat istog koda u web pregledniku.

3.4. JSON

JSON (*JavaScript Object Notation*) je jednostan format za spremanje i razmjenu podatka. JSON koristi sintaksu programskog jezika JavaScript, ali se može koristiti od strane bilo kojeg programskog jezika. [9]

JSON format je sintaksom identičan kodu za kreiranje JavaScript objekta. Zbog sličnosti sa JavaScript sintaksom u JavaScript programima možemo koristit standardne JavaScript funkcije za pretvaranje JSON podataka u nativne JavaScript objekte. Podaci u JSON sintaksi dolaze u paru ime/vrijednost. Podaci su razdvojeni zarezima. Vitičaste zagrade sadrže objekte, dok uglate zagrade sadrže nizove.

Kod 3.2. Primjer JSON sintakse

```
"auto":"BMW"
```

U kodu 3.2. naveden primjer JSON koda koji opisuje JSON podatak pisan u paru ime/vrijednost. Ime i vrijednost moraju biti pod znacima navodnika, svaki posebno te između njih mora biti dvotočka. JSON vrijednosti mogu biti sljedeći tipovi podataka:

- broj (cijeli ili decimalni),
- string (pod znacima navodnika),
- boolean (istina ili laž),

- niz (u uglatim zagradama),
- objekt (u vitičastim zagradama),
- null.

Kod 3.3. Primjer JSON objekt

```
{"ime":"Marko", "prezime":"Markić"}
```

U kodu 3.3. naveden primjer JSON koda koji opisuje JSON objekt. JSON objekti mogu sadržavati više ime/vrijednost parova. JSON nizovi također mogu sadržavati više ime/vrijednost parova što je prikazano u sljedećem primjeru pod kod 3.4.

Kod 3.4. Primjer JSON polja

```
"zaposlenici":
[
{"ime":"Tomislav","prezime":"Tomić"},
{"ime":"Tihana","prezime":"Mišević-Kujundžić"},
{"ime":"Petar","prezime":"Petrović"}
]
```

Datoteka koja sadrži JSON ima ekstenziju „.json“. JSON se često koristi za čitanje podataka sa web poslužitelja i prikazvanje podataka na web stranici kako bi podaci bili dostupni programerima.

3.5. GeoJSON

GeoJSON je format za kodiranje raznih geografskih podatkovnih struktura. GeoJSON objekt može predstavljati geometriju, svojstvo, ili skupinu svojstava. GeoJSON podržava sljedeće geometrijske tipove:

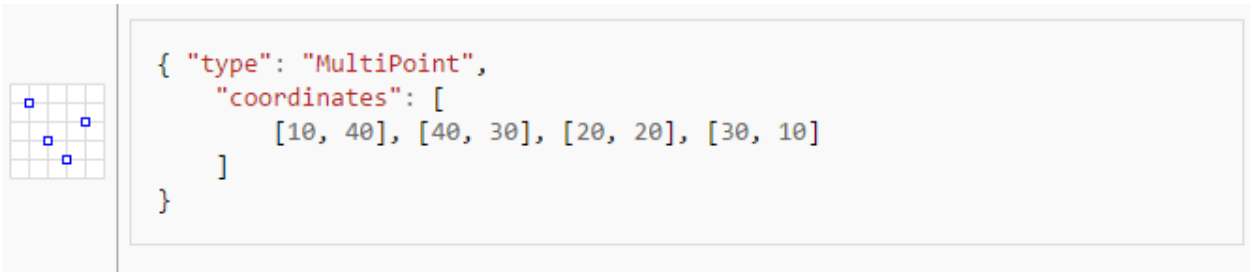
- točka,
- pravac,
- mnogokut,
- više točaka,
- više pravaca,
- više mnogokuta,
- geometrijska kolekcija. [10]

Svojstva u GeoJSON-u sadrže geometrijski objekt i dodatna svojstva. Cjelovita GeoJSON podatkovna struktura je uvijek objekt. U GeoJSON-u objekt se sastoji od skupine ime/vrijednost parova kao što je opisano u prethodnom potpoglavlju. GeoJSON se uvijek sastoji od jednog objekta koji ima sljedeća svojstva:

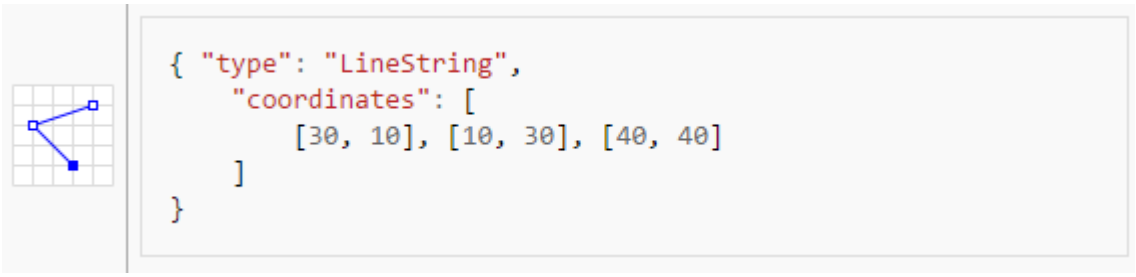
- GeoJSON objekt može imati neograničen broj članova(ime/vrijednost),
- GeoJSON objekt mora sadržavati član sa imenom *„type“* ,
- vrijednost člana pod imenom *„type“* mora biti jedan od sljedećih*: „Point“*, *„Multipoint“*, *„LineString“*, *„MultiLineString“*, *„Polygon“*, *„MultiPolygon“*, *„GeometryCollection“*, *„Feature“* ili *„FeatureCollection.“* ,
- GeoJSON objekt može imati izborni „*crs*“ član te „*bbox*“ član. [10]

GeoJSON geometrijski objekt bilo kojeg tipa osim *„GeometryCollection“* mora sadržavati član s imenom *„coordinates"*. Vrijednost člana „*coordinates*“ je uvijek niz. Struktura elemenata u nizu ovisi o geomerijskom tipu. Za tip „*Point*“ vrijednost člana „*coordinates*“ mora biti jedna točka. Za tip „*Multipoint*“ vrijednost člana „*coordinates*“ mora biti niz točaka. Za tip „*LineString*“ vrijednost člana „*coordinates*“ mora biti niz sa dvije ili više točke. Za tip „*MultiLineString*“ vrijednost člana „*coordinates*“ mora biti niz od više „*LineString*“ nizova. Za tip „*Polygon*“ vrijednost člana „*coordinates*“ mora biti niz od nizova Linearnih Prstenova (engl. *LinearRing*). Linearni prsten je zatvoreni „*LineString*“ sa 4 ili više točke gdje su prva i posljednja točka jednake. Za poligone s više linearnih prstenova prvi mora biti vanjski prsten, dok su ostali unutar njega. Za tip „*MultiPolygon*“ vrijednost člana „*coordinates*“ mora biti niz od nizova poligona. [10]

Slika 3.3. Primjer GeoJSON objekta za tip „*Point*“ [Izvor: https://en.wikipedia.org/wiki/GeoJSON]

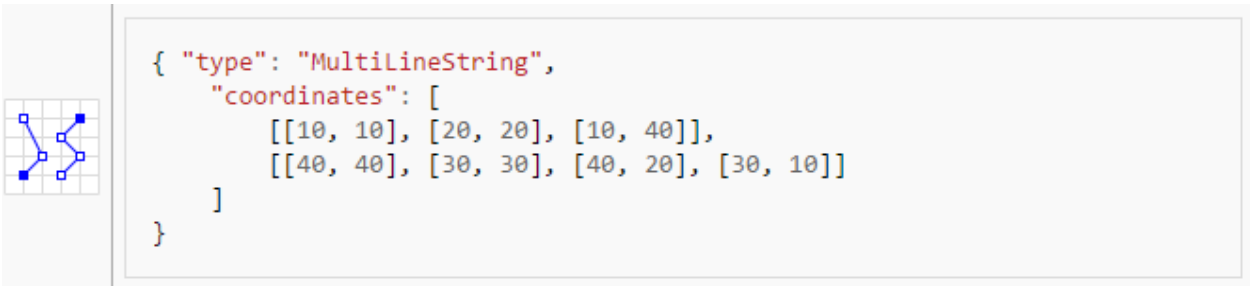


Slika 3.4. Primjer GeoJSON objekta za tip „*MultiPoint*“ [Izvor: https://en.wikipedia.org/wiki/GeoJSON]



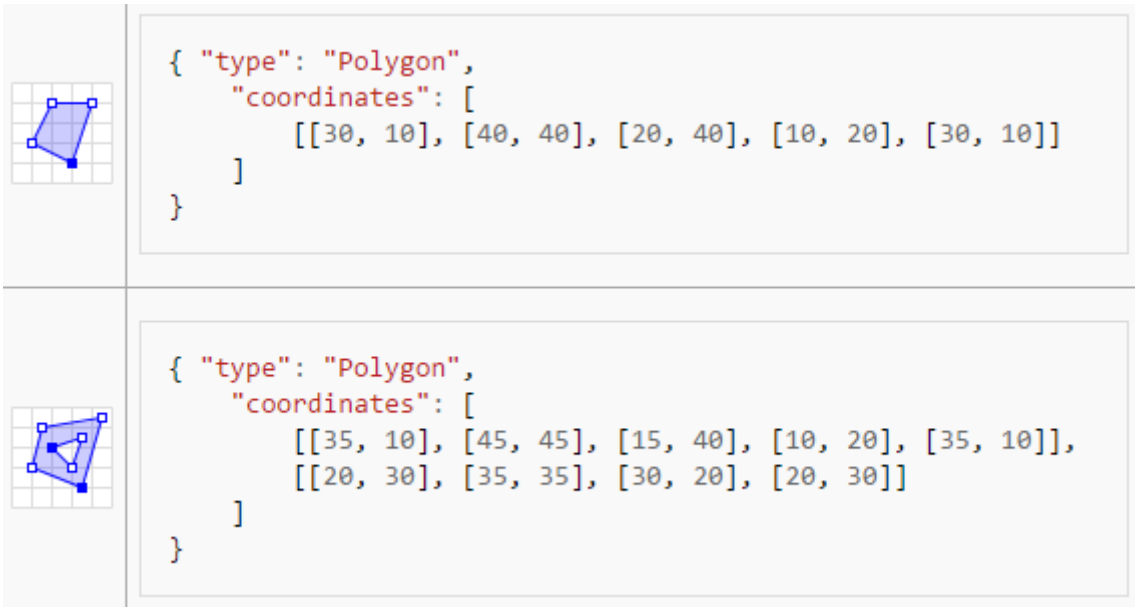
```
{ "type": "LineString",
    "coordinates": [
        [30, 10], [10, 30], [40, 40]
    ]
}
```

Slika 3.5. Primjer GeoJSON objekta za tip „*LineString*“ [Izvor: https://en.wikipedia.org/wiki/GeoJSON]



```
{ "type": "MultiLineString",
    "coordinates": [
        [[10, 10], [20, 20], [10, 40]],
        [[40, 40], [30, 30], [40, 20], [30, 10]]
    ]
}
```

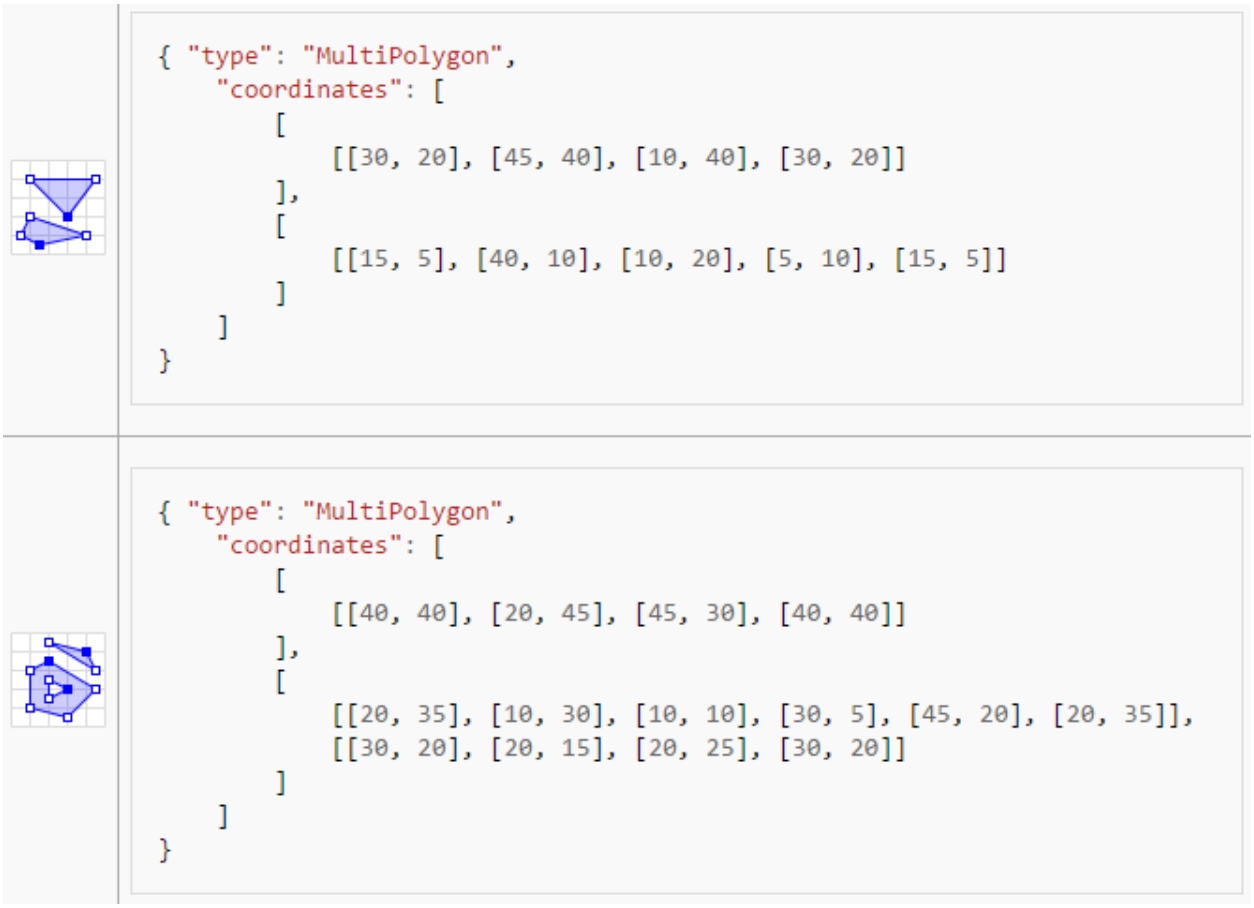
Slika 3.6. Primjer GeoJSON objekta za tip „*MultiLineString*“ [Izvor: https://en.wikipedia.org/wiki/GeoJSON]



```
{ "type": "Polygon",
    "coordinates": [
        [[30, 10], [40, 40], [20, 40], [10, 20], [30, 10]]
    ]
}
```

```
{ "type": "Polygon",
    "coordinates": [
        [[35, 10], [45, 45], [15, 40], [10, 20], [35, 10]],
        [[20, 30], [35, 35], [30, 20], [20, 30]]
    ]
}
```

Slika 3.7. Primjer GeoJSON objekta za tip „*Polygon*“ [Izvor: https://en.wikipedia.org/wiki/GeoJSON]



```
{ "type": "MultiPolygon",
    "coordinates": [
        [
            [[30, 20], [45, 40], [10, 40], [30, 20]]
        ],
        [
            [[15, 5], [40, 10], [10, 20], [5, 10], [15, 5]]
        ]
    ]
}
```

```
{ "type": "MultiPolygon",
    "coordinates": [
        [
            [[40, 40], [20, 45], [45, 30], [40, 40]]
        ],
        [
            [[20, 35], [10, 30], [10, 10], [30, 5], [45, 20], [20, 35]],
            [[30, 20], [20, 15], [20, 25], [30, 20]]
        ]
    ]
}
```

Slika 3.8. Primjer GeoJSON objekta za tip „*MultiPolygon*“ [Izvor: https://en.wikipedia.org/wiki/GeoJSON]

Na slikama 3.3., 3.4., 3.5., 3.6., 3.7. i 3.8. navedeni su jednostavni primjeri GeoJSON objekata za sve vrste tipova.

3.6. Leaflet

Leaflet je JavaScript biblioteka koja omogućava kreiranje web aplikacija za prikazivanje geografskih karata. Prva verzija LefletJS je objavljena 2011. godine, a podržava većinu mobilnih i desktop platformi. [11]

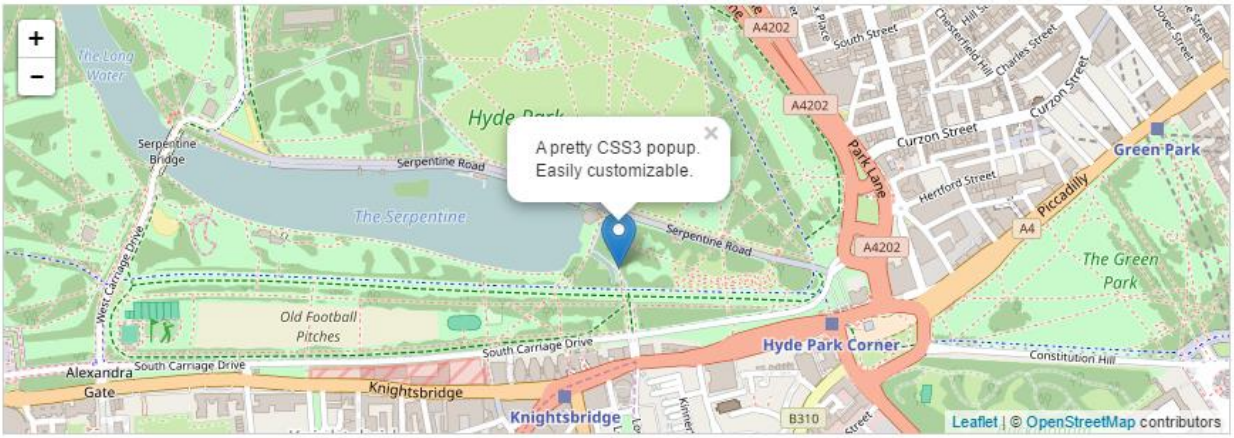
Uz *OpenLayers* i *GoogleMaps API* je jedna od najpopularnijih JavaScript biblioteka za izradu navedenog tipa web aplikacija. Leaflet omogućava razvijateljima web aplikacija kreiranje web karata bez poznavanja GIS-a (*Geografski informacijski sustav*). Omogućava korištenje podataka iz GeoJSON datoteka, dodavanje stilova, dodavanje markera na kartu, dodavanje skočnih prozora itd. Leaflet izvorno podržava korištenje WMS (*Web Map Service*) slojeva, GeoJson slojeva,

vektorskih slojeva te *Tile* slojeva. Drugi tip slojeva moguće je koristiti umetanjem dodataka. Glavni tipovi Leaflet objekata su rasterskog tipa (*TileLayer* i *ImageOverlay*), vektorskog tipa (*Path*, *Polygon*, *Circle* itd.), grupiranog tipa (*LayerGroup*, *FeatureGroup* i *GeoJSON*), kontrole (*Zoom*, *Layers* itd.). Također, postoje razne korisne klase za upravljanje izbočenjima, transformacijama te klase za interakciju sa DOM elementima. Leaflet podržava sve novije verzije web preglednika. [11]

Leaflet je izravno usporediv sa OpenLayers bibliotekom jer su obje JavaScript biblioteke otvorenog koda te se izvode na strani klijenta. Leaflet biblioteka ima mnogo manje koda te je baza koda novija. Koristi najnovije značajke Javascript jezika te HTML5 i CSS3. OpenLayers podržava WFS (*Web Feature Sevice*) dok Leaflet ne podržava. Leaflet također ima slične mogućnosti kao i biblioteke Google Maps API te Bind Maps API. U radu je korišten Leaflet zbog jednostavnosti te zbog velike podrške programerske zajednice. U kodu 3.5. je naveden jednostavni primjer sa službene Leflet web stranice te je prikazan rezultat koda u web pregledniku na slici 3.9.

Kod 3.5. Primjer postavljanja karte

```
var map = L.map('map').setView([51.505, -0.09], 13);

L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {
    attribution: '&copy; <a href="http://osm.org/copyright">OpenStreetMap</a>
contributors'
}).addTo(map);

L.marker([51.5, -0.09]).addTo(map)
    .bindPopup('A pretty CSS3 popup.<br> Easily customizable.')
    .openPopup();
```



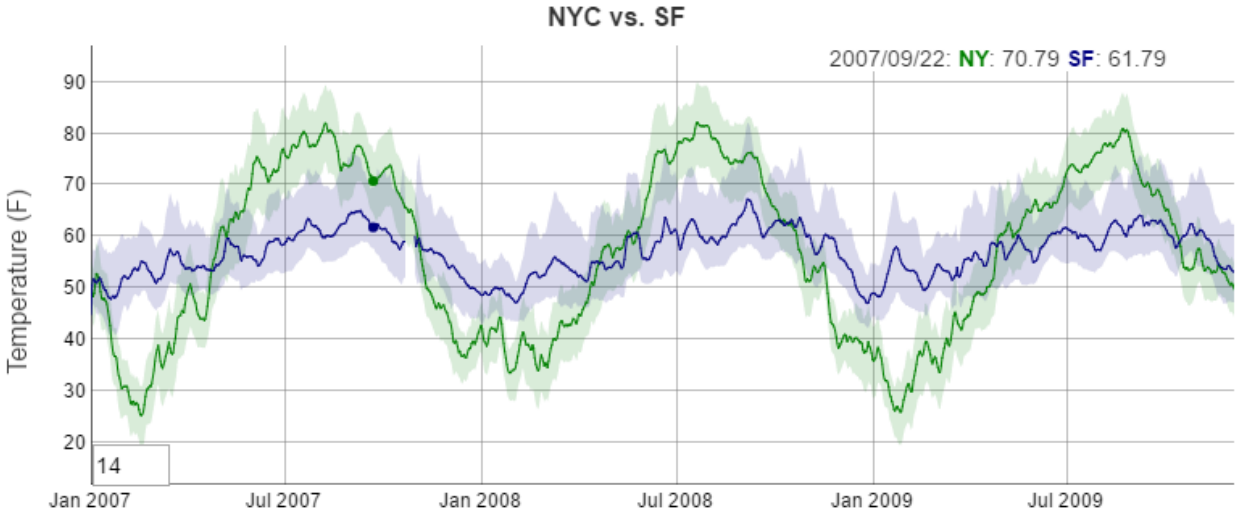
Slika 3.9. Rezultat navedenog koda 3.5. [Izvor: autor]

3.7. Dygraphs

Dygraphs je JavaScript biblioteka koja omogućava i olakšava kreiranje grafova. Može rukovati većom količinom podataka, graf je interaktivan te je visoko podesiv. Radi na najpopularnijim web preglednicima. [12]

Kod 3.6. Postavljanje linijskog grafa

```
new Dygraph(div, "ny-vs-sf.txt", {
  legend: 'always',
  title: 'NYC vs. SF',
  showRoller: true,
  rollPeriod: 14,
  customBars: true,
  ylabel: 'Temperature (F)',
});
```



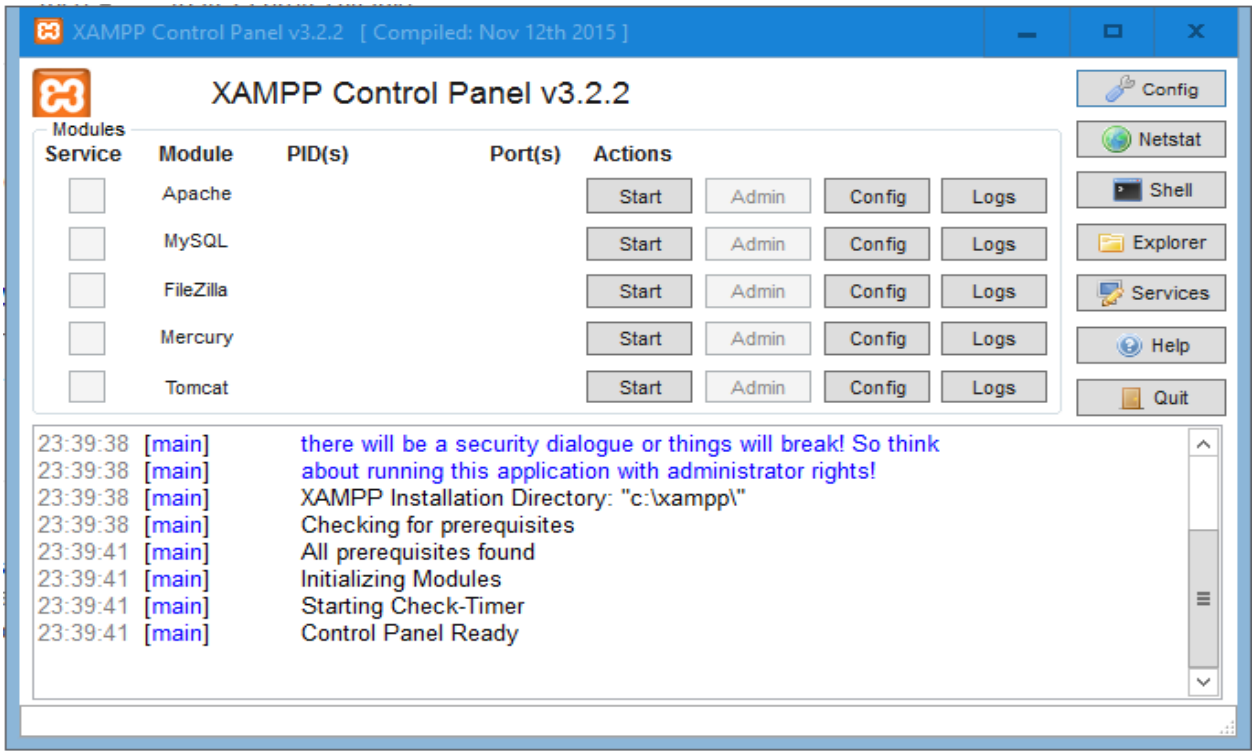
Slika 3.10. Rezultat navedenog koda pod Kod 3.6. [Izvor: autor]

Navedeni kod 3.6. predstavlja primjer sa službene stranice te je navedena slika 3.10. koja predstavlja rezultat istog koda u web pregledniku.

3.8. XAMPP

XAMPP (*X Apache MySQL PHP Perl*) je poslužiteljski paket koji je namjenjen za jednostavnu instalaciju Apache poslužitelja na računala. Otvorenog je koda te ga je moguće koristiti na različitim platformama. [13]

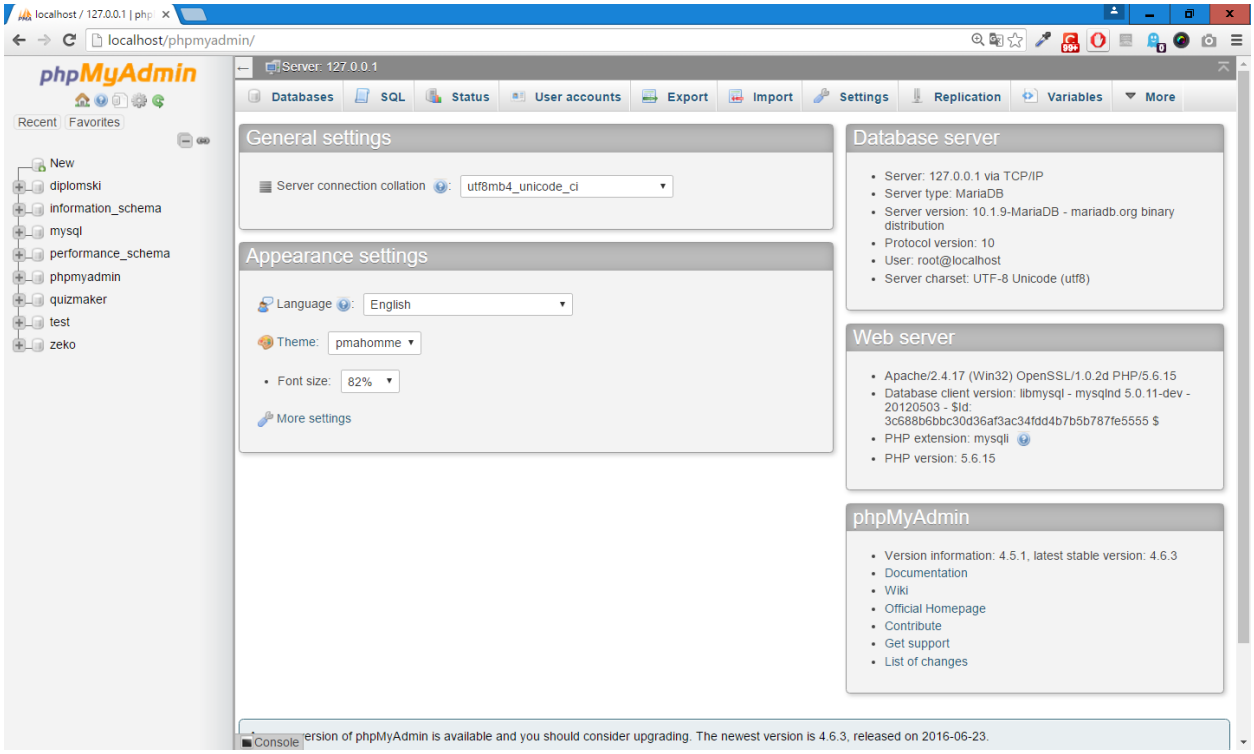
Osnovni dijelovi paketa su Apache poslužitelj i MySQL baza podataka. Ima mogućnosti interpretirati PHP i PERL skripte. Uz novije verzije XAMMP-a također dolazi i phpMyAdmin web aplikacija koja će biti opisana u slijedećem potpoglavlju. Namijenjen je kao pomagalo pri izradi dinamičnih web aplikacija. Prije nekoliko godina bilo je potrebno zasebno instalirati PHP, MySQL i Apache, a instalacijom XAMPP-a dobiva se sve odjednom. Na slici 3.11. je prikazana kontrolna ploča XAMPPA-a.



Slika 3.11. Kontrolna ploča XAMPP-a [Izvor: autor]

3.9. PhpMyAdmin

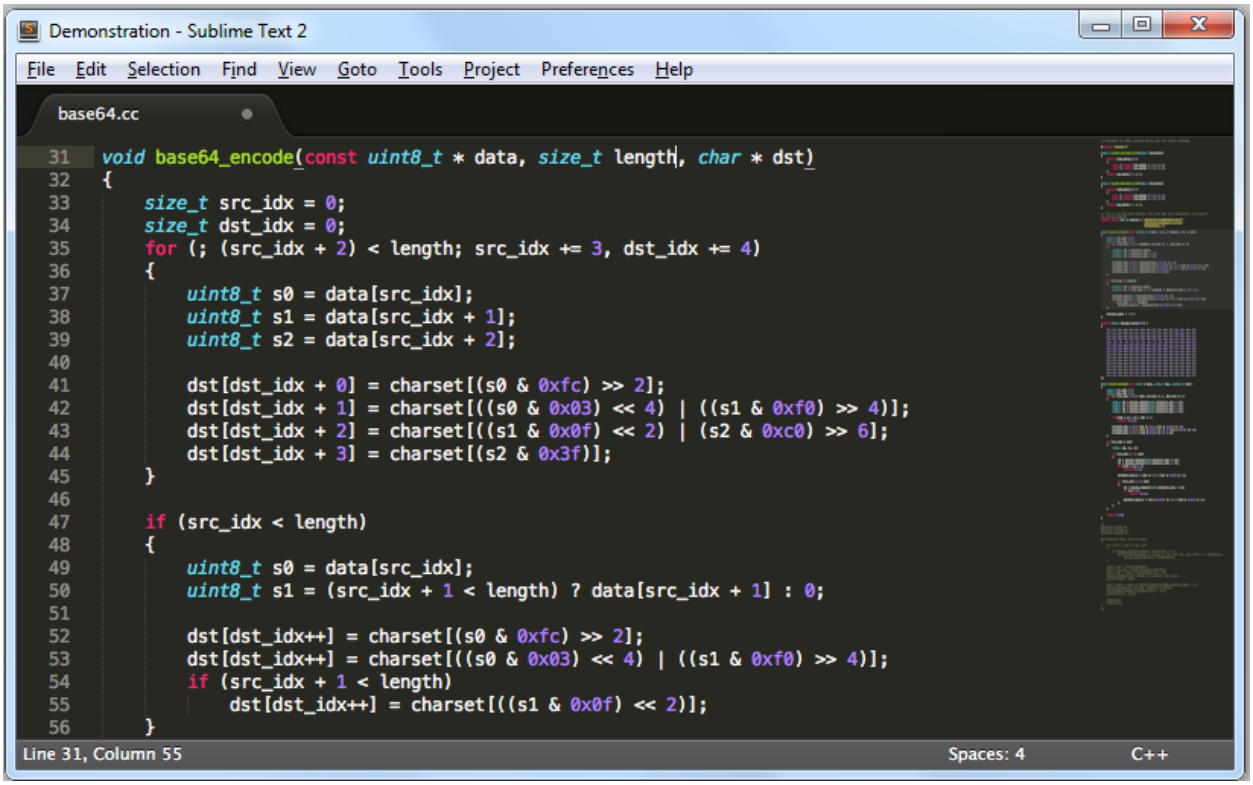
PhpMyAdmin je besplatna web aplikacija koja služi za upravljanje i administraciju bazom podataka, a dolazi u XAMPP instalacijskom paketu. Aplikacija je pisana u PHP-u. PhpMyAdminu pristupamo preko linka http://localhost/phpmyadmin. PhpMyAdmin omogućava upravljanje bazom podataka, tablicama, korisnicima, poljima, dozvolama itd. Također omogućava izvršavanje upita na bazu preko SQL jezika. Grafičko sučelje je prevedeno na 72 jezika. Na slici 3.12. je prikazano korisničko sučelje u web pregledniku.[14]



Slika 3.12. PhpMyAdmin korisničko sučelje u lokalnom okruženju [Izvor: autor]

3.10. Sublime Text

Za pisanje aplikacije korišten je SublimeText uređivač teksta. Sublime Text je pisan u Pythonu i moguće ga je koristiti na svim popularnim operacijskim sustavima. Podržava mnogo jezika, a funkcionalnosti se mogu proširivati instalacijom dodataka od strane korisnika. Na slici 3.13. je prikazano korisničko sučelje navedenog uređivača teksta.



Slika 3.13. Korisničko sučelje SublimeText uređivača teksta [Izvor: autor]

4.IMPLEMENTACIJA I PRIKAZ APLIKACIJE

U ovom poglavlju prikazana je implementacija aplikacije za grafičko prikazivanje podataka prikupljenih senzorskom mrežom. Također su prikazani rezultati izrade aplikacije. Za izradu aplikacije koristile su se sljedeće programske tehnologije: HTML, CSS, JavaScript, PHP te baza podataka MySQL. Sve tehnologije su otvorenog koda. Također, koristili su se dodaci koji višestruko ubrzavaju razvoj web aplikacije. Korišteni su programski okviri Bootstrap, jQuery, Date Range Picker, JSON, GeoJSON, Leaflet i DyGraphs. Od alata je korišten poslužiteljski paket XAMPP, za rad sa bazom podataka aplikacija phpMyAdmin te za pisanje samog koda SublimeText uređivač teksta. Implementacija se može podijeliti u tri dijela. U prvom dijelu se radilo sa bazom, kreirana je tablica te je popunjena podacima. Drugi dio se odnosi na izradu grafičkog prikazivanja podataka na geografskoj karti Republike Hrvatske. Izradu mape omogućio je dodatak Leaflet. Treći dio je odrađen pomoću dodatka DyGraphs, a odnosi se na grafički prikaz podataka na linijskom grafu.

4.1. Rad sa bazom podataka

Za izradu ove aplikacije korišten je besplatan sustav za upravljanje bazom podataka MySQL. MySQL baza je relacijskog tipa i moguće ju je koristiti u kombinaciji s PHP-om. Za kreiranje baze i rad s bazom koristila se web aplikacija phpMyAdmin. Za potrebu ove aplikacije kreirana je baza s jednom tablicom, gdje će biti spremljeni podaci koji predstavljaju broj stanice, datum te parametre. Parametri su vlažnost tla (koji se prikupljaju senzorima na šest različitih dubina), temperatura zraka, vlažnost zraka, brzina vjetra i količina padalina.

1	**stanica_id**	int(2)
2	**datum**	date
3	**sm1**	float
4	**sm2**	float
5	**sm3**	float
6	**sm4**	float
7	**sm5**	float
8	**sm6**	float
9	**at1**	float
10	**ah1**	float
11	**ws1**	float
12	**pp1**	float

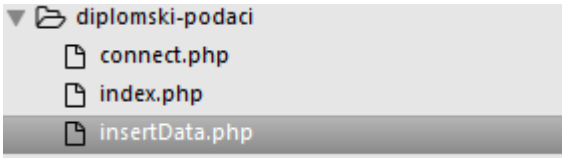
Slika 4.1. Struktura tablice „podaci“ [Izvor: autor]

Na slici 4.1. prikazana je struktura tablice „podaci“. Iz slike je vidljivo da se tablica sastoji od 12 stupaca. Prvi stupac je *stanica_id*, a predstavlja broj stanice. Prva stanica je tipa *int* što znači da se u taj stupac mogu spremiti samo cijeli brojevi, a broj 2 označava maksimalan broj znamenki broja, što je za potrebu ove aplikacije dovoljno. Stupac *datum* predstavlja datum i vrijeme mjerenja te je tipa *datetime*. Ostali stupci predstavljaju mjerene parametre. Nazivi su inicijali parametara na engleskom jeziku: vlažnost tla - sm (eng. *soil moisture*), vlažnost zraka – ah1 (eng. *air humiditiy*), temperatura zraka - at1 (eng. *air temperature*), padaline - pp1 (eng. *precipatition*) te brzina vjetra - ws1 (eng. *wind speed*). Stupci koji predstavljaju parametre su tipa *float*, što znači da se u taj stupac mogu spremiti decimalni brojevi.

stanica_id	datum	sm1	sm2	sm3	sm4	sm5	sm6	at1	ah1	ws1	pp1
1	2016-01-10 08:00:00	15.8	35.3	16	52.8	14	39.8	-7.2	39.3	6.6	0
1	2016-01-10 09:00:00	15.2	35.6	15.6	52.8	15	39.4	-7.8	39	6.3	0
1	2016-01-10 10:00:00	16	35.5	15	52.2	14.2	39.9	-7	40	6.4	0
1	2016-01-10 11:00:00	16	35.5	15.7	52.4	14.6	39.5	-7	39.9	6.7	0
1	2016-01-10 12:00:00	15	35	15.6	52.9	14.8	40	-8	39	6.7	0
1	2016-01-10 13:00:00	15.3	35.9	15.3	52.4	14.5	39.5	-7.1	39.4	6.7	0
1	2016-01-10 14:00:00	15.9	35.1	15.7	52.9	14.2	39.8	-7.7	39.8	6.4	0
1	2016-01-10 15:00:00	15.2	35.2	15.8	52.7	15	39.3	-7	39.9	7	0
1	2016-01-10 16:00:00	15.9	35.8	15.5	52.4	14.1	39.8	-7.9	40	6.1	0
1	2016-01-10 17:00:00	15.5	35.7	15.3	53	14.3	39.2	-7.6	39.8	6.7	0
1	2016-01-10 18:00:00	15.9	35.8	15.9	53	14.8	39.7	-8	39.4	6.9	0
1	2016-01-10 19:00:00	15.1	35.7	15.7	52.6	14	39.2	-7.3	39.8	6.8	0
1	2016-01-10 20:00:00	15.8	35.4	15.8	52.5	14.2	39.9	-7.1	39.6	6.5	0
1	2016-01-10 21:00:00	15.5	35.7	15.5	52.3	14.4	39.9	-7.9	39.6	6.1	0
1	2016-01-10 22:00:00	15.4	35	15.3	52.6	14.7	39.7	-7.6	39.9	6	0
1	2016-01-10 23:00:00	15	35.1	15.1	52.9	14	39	-7.6	39.6	6.8	0
1	2016-01-11 00:00:00	78	37	53	80	52	39	12	22	6	0
1	2016-01-11 01:00:00	78.6	37.2	53	80.3	52.3	39	12	22.5	6.4	0
1	2016-01-11 02:00:00	78.5	37.1	53.8	80.8	52.3	40	12.3	22.2	6.1	0
1	2016-01-11 03:00:00	78.3	37.1	53.9	80.1	52.3	40	12.4	22.6	6.6	0
1	2016-01-11 04:00:00	78.5	37.7	53.5	80.2	52.7	40	12.3	22.5	6.9	0
1	2016-01-11 05:00:00	78.4	37.1	53.7	81	52.2	39.5	12.2	23	6.4	0
1	2016-01-11 06:00:00	78.1	37.4	53.3	80.8	52.8	39.4	12.5	22	6.4	0
1	2016-01-11 07:00:00	78.9	37.3	53	80.1	53	39.1	12.4	22.8	6.9	0
1	2016-01-11 08:00:00	78	37.9	53.1	80.6	52	39.7	12.6	22.2	6.9	0

Slika 4.2. Prikaz ispunjene tablice „podaci“ [Izvor: autor]

Slika 4.2. prikazuje popunjenu tablicu s podacima. Tablica je popunjena s podacima za 365 dana. Podaci predstavljaju očekivane vrijednosti te su generirani uz pomoć PHP skripte. Svaki dan ima 24 mjerenja s razmakom od sat vremena. Stupci *sm1, sm2, sm3, sm4, sm5, sm6* te *ah1* izraženi su u postocima te mogu imati vrijednost od 0-100. Vrijednost stupca *at1* izražena je u celzijevim stupnjevima, a može imati vrijednost od -20 do 40. Vrijednost stupca *ws1* je izražena u km/h, a može imati vrijednost od 0 do 40. Vrijednost stupca *pp1* izražena je u mm te može imati vrijednost od 0-100.



Slika 4.3. Struktura direktorija diplomski-podaci [Izvor: autor]

Na prethodnoj slici 4.3. prikazane su datoteke pomoću kojih su generirani i umetnuti podaci u bazu. U datoteci *connect.php* pisan je kod koji omogućava spajanje na bazu. U datoteci *index.php* se nalazi kod koji kreira podatke te se njemu poziva funkcija *insertData()* koja je smještena u datoteci *insertData.php*. Podaci se generiraju korištenjem nativne PHP funkcije *rand()*. Kreira se skup podataka za 350 dana koristeć *for* petlju.

Kod 4.1. Spajanje na MySQL poslužitelj

```
<?php
// Connect to MySQL
$mysqli = new mysqli( "localhost", "root", "", "diplomski");
// Check our connection
if ( $mysqli->connect_error ) {
die( 'Connect Error: ' . $mysqli->connect_errno . ': ' . $mysqli-
>connect_error );
}
?>
```

Kod 4.1. je smješten u datoteci *connect.php*. Kod predstavlja objektno orijentirani PHP način spajanja na MySQL poslužitelj. U prvom redu kreira se nova instanca klase *mysqli* . Konstruktor *mysqli::__construct* u ovom primjeru prima za parametre ime poslužitelja, korisničko ime, zaporku te ime baze.

Kreiran je objekt *$mysqli* koji je predstavlja vezu na MySQL posužitelj. U narednom dijelu koda se poziva metoda *$mysqli->connect_error* koja vraća *true* ili *false*. U slučaju da metoda vrati *true* znači da je došlo do greške sa spajanjem na poslužitelj te se izvršava dio koda gdje se pozivaju metode *$mysqli->connect_errno* i *$mysqli->connect_error* koje ispisuju broj i vrstu greške.

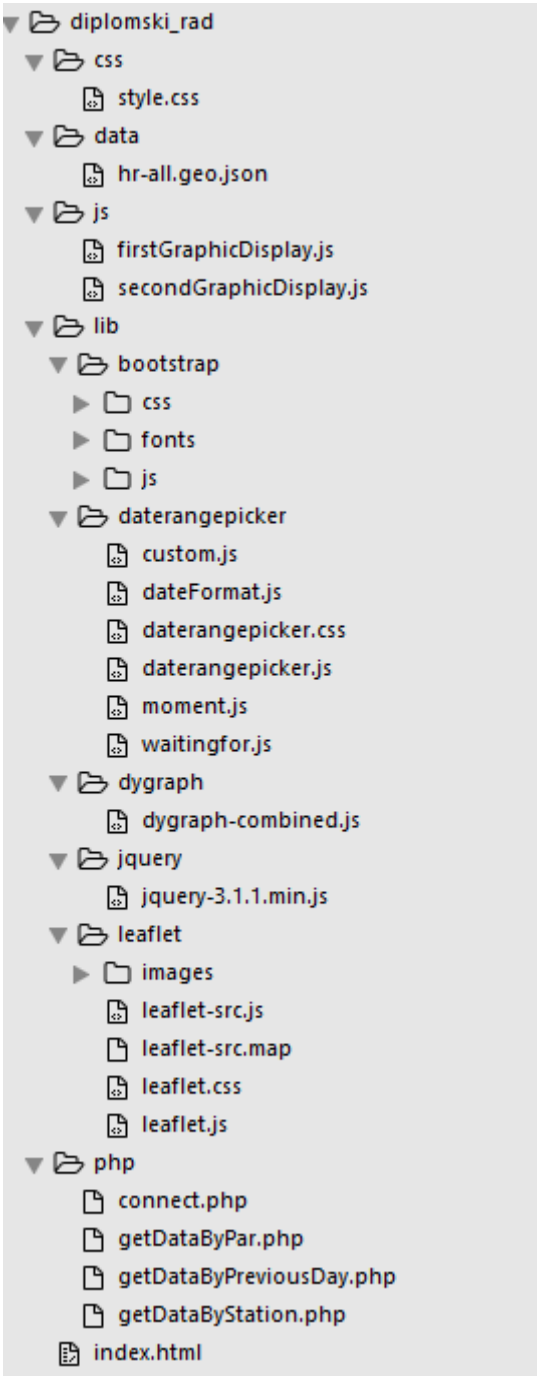
Kod 4.2. Kreiranje parametara

```
$sm1base = rand(0, 100);
$sm2base = rand(0, 100);
$sm3base = rand(0, 100);
$sm4base = rand(0, 100);
$sm5base = rand(0, 100);
$sm6base = rand(0, 100);
$ah1base = rand(0, 100);
$at1base = rand(-20, 40);
$pp1base = rand(0, 100);
$ws1base = rand(0, 40);
```

Kod 4.2. predstavlja kreiranje varijabli mjernih parametara koristeći funkciju *rand()* koja prima dva parametra. Prvi parametar predstavlja najmanju vrijednost parametra dok drugi parametar predstavlja najveću moguću vrijednost parametra.

4.2. Opis datoteka aplikacija

U ovom dijelu diplomskog rada bit će navedene i ukratko opisane sve datoteke izrađene aplikacije. Dio datoteka se odnosi na programske okvire koji su se koristili pri izradi aplikacije. Sve datoteke programskih okvira su preuzete sa službenih stranica kreatora programskih okvira. Drugi dio datoteka je kreiran pri izradi aplikacije.



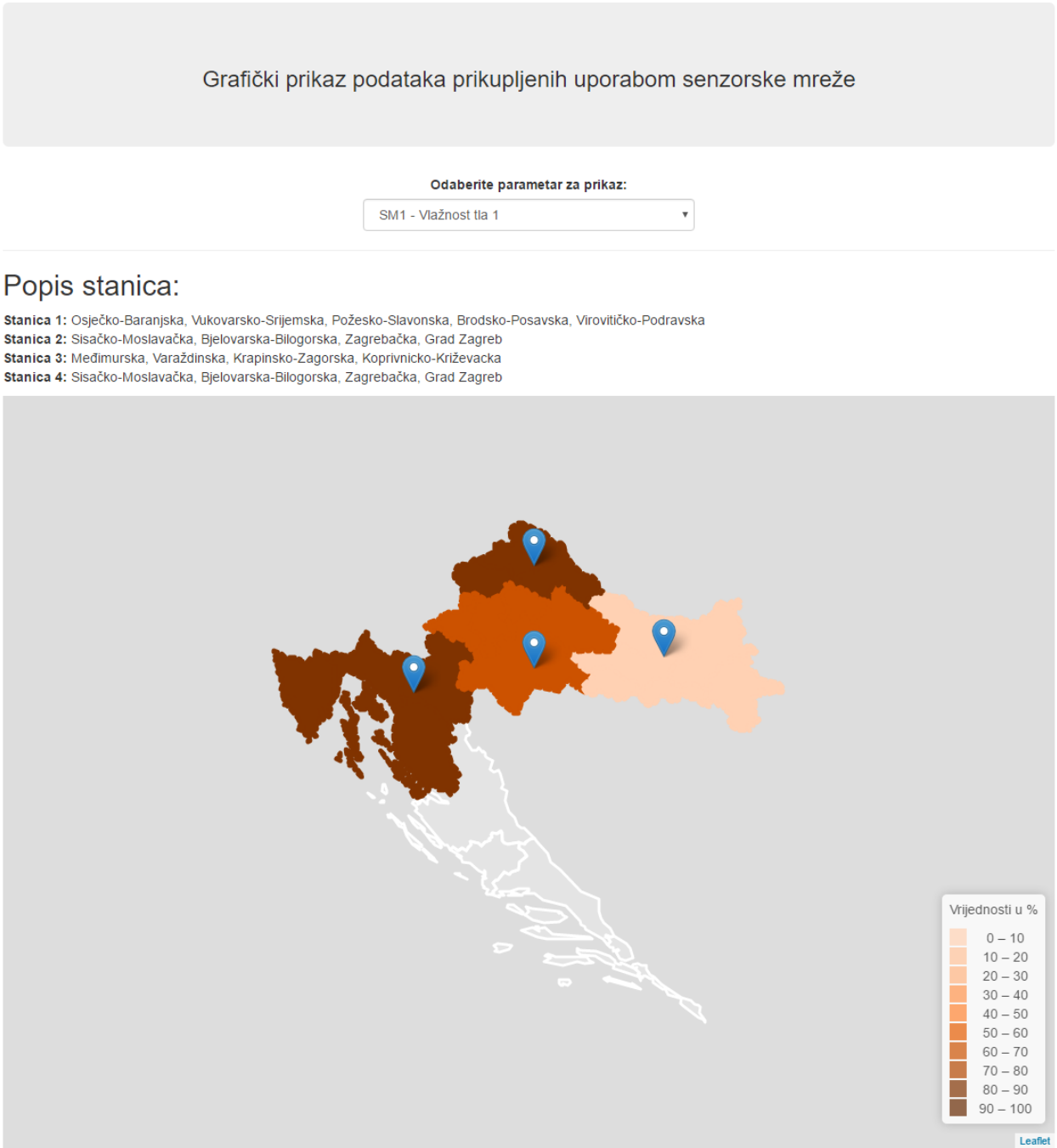
Slika 4.4. Struktura direktorija „diplomski_podaci“ [Izvor: autor]

Na slici 4.4. prikazana je struktura direktorija *diplomski_podaci*. U navedenom direktoriju su smještene sve datoteke koje aplikacija koristi. U direktoriju *diplomski_podaci* se nalaze novi direktoriji *css, data, js, lib, php* te HTML datoteka *index.html*. U direktoriju naziva *css* se nalazi datoteka *style.css* u kojoj su pisani dodatni stilovi aplikacije. Sljedeći direktorij je *data* u kojem se nalazi datoteka *hr-all.geo.json* u kojoj je smješten GeoJSON kod koji sadrži informacije za kartu Hrvatske. Datoteka *hr-all.geo.json* je preuzeta s web stranice

http://alas.matf.bg.ac.rs/~mi09109/hrv_regional.geojson. Direktorij *js* sadrži dvije JavaScript datoteke *firstGraphicDisplay.js* i *secondGraphicDisplay.js*. U datoteci *firstGraphicDisplay.js* je smješten kod pisan za grafički prikaz podataka na karti Hrvatske, dok druga datoteka *secondGraphicDisplay.js* sadrži kod koji se odnosi na grafički prikaz podataka na linijskom grafu. Direktorij *lib* sadrži datotetke programskih okvira korištenih u aplikaciji raspoređenih u pet poddirektorija: *bootstrap*, *daterangepicker*, *dygraph*, *jquery* i *leaflet*. Svi programski okviri su besplatni te su preuzete njihove najnovije verzije sa službenih web stranica. Zadnji direktorij pod nazivom *php* sadrži četiri PHP skripte pod nazivima *connect.php*, *getDataByPar.php*, *getDataByPreviousDay.php* i *getDataByStation.php*. U prvoj skripti je kreiran objekt koji predstavlja povezivanje na MySQL bazu. U preostale tri skripte je smješten kod u kojem se rade upiti na bazu pomoću prethodno navedenog objekta, a kao rezultat ispisuju JSON strukturu podataka koji se prikazuju u aplikaciji. Jedina datoteka koju sadrži direktorij *diplomski_podaci* je *index.html*. Kada korisnik otvori direktorij *diplomski_podaci* putem web preglednika učitava mu se datoteka *index.html* u kojoj su uključeni svi programski okviri te JavaScript datoteke koje se odnose na logiku aplikacije.

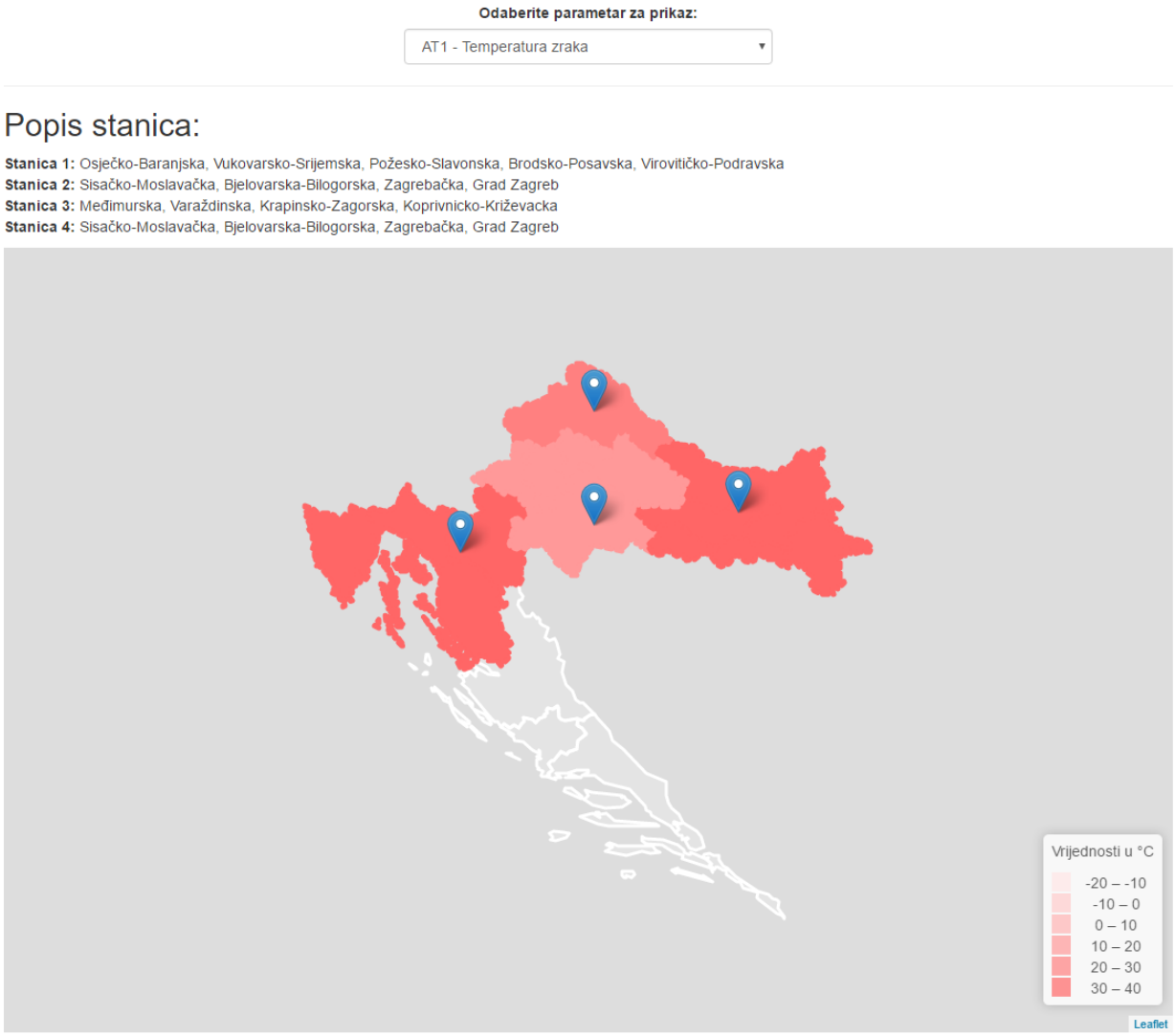
4.3. Prikaz podataka na karti Hrvatske

U ovom dijelu rada prvo će biti prikazan rezultat izrade prvog dijela aplikacije koji predstavlja grafički i tablični prikaz podataka na karti Hrvatske.



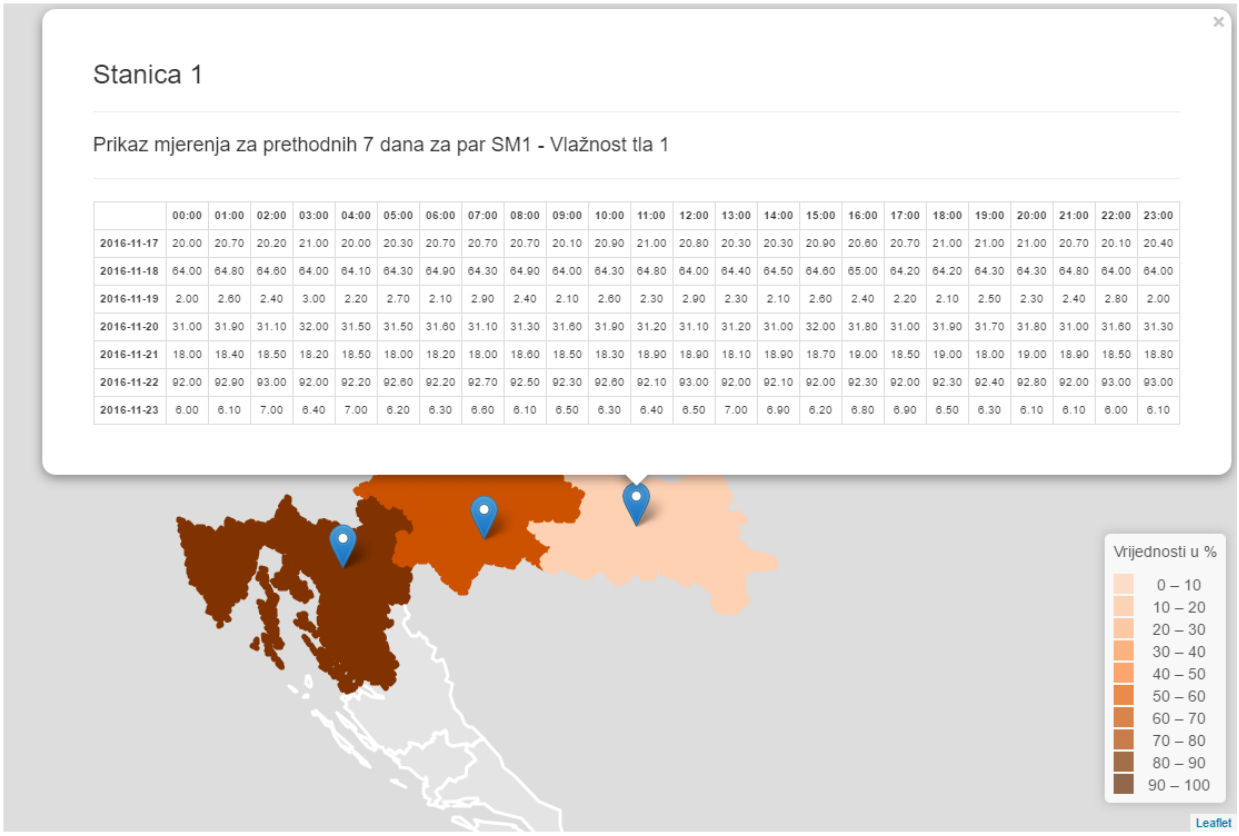
Slika 4.5. Grafički prikaz podataka za parametar Vlažnost tla 1 [Izvor: autor]

Slika 4.5. predstavlja dio aplikacije koji omogućuje grafički i tablični prikaz podataka na karti Hrvatske. Na karti su dodana četiri markera koji označavaju stanice. Svaka stanica obuhvaća jedno područje. Stanica 1 obuhvaća sljedeće županije: Osječko-Baranjska, Vukovarsko-Srijemska, Požesko-Slavonska, Brodsko-Posavska, Virovitičko-Podravska. Stanica 2 obuhvaća: Sisačko-Moslavačka, Bjelovarska-Bilogorska, Zagrebačka, Grad Zagreb. Stanica 3 obuhvaća: Međimurska, Varaždinska, Krapinsko-Zagorska, Koprivnicko-Križevacka. Stanica 4 obuhvaća: Sisačko-Moslavačka, Bjelovarska-Bilogorska, Zagrebačka te Grad Zagreb. Iznad karte vidljiva je mogućnost odabira parametra koji se želi prikazati. Za svaku stanicu i odabrani parametar se povlače podaci iz baze za prethodni dan. Svaka stanica za prehodni dan sadrži 24 mjerenja svakog parametra. Na temelju 24 mjerenja se izračunava prosječna vrijednost za svaku od stanica te na temelju te vrijednosti boja se područje određenom nijansom smeđe boje. U desnom donjem kutu slike nalazi se legenda koja predstavlja nijanse smeđe boje kojima se karta boja. U ovom slučaju je odabran parametar koji predstavlja vlažnost tla čije su vrijednosti izražene u postocima od 0 do 100. Za sve parametre koji se odnose na vlažnost tla uzimaju se nijanse smeđe boje. Za vlažnost zraka, brzinu vjetra i padaline se uzimaju nijanse plave boje dok se za temperaturu zraka uzimaju nijanse crvene boje.



Slika 4.6. Grafički prikaz podataka za parametar Temperatura zraka [Izvor: autor]

Promjenom parametra na padajućem izborniku automatski se mijenja i prikaz na karti. Na slici 4.6. je vidljivo da je odabran parametar temperatura zraka. Za temperaturu zraka je predodređena crvena boja. Na legendi se vidi da je vrijednost temperature izražena u celzijevim stupnjevima. Klikom na jednu od stanica otvara se tablični prikaz podataka kao što je prikazano na slici 4.7.



	00:00	01:00	02:00	03:00	04:00	05:00	06:00	07:00	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00
2016-11-17	20.00	20.70	20.20	21.00	20.00	20.30	20.70	20.70	20.70	20.10	20.90	21.00	20.80	20.30	20.30	20.90	20.60	20.70	21.00	21.00	21.00	20.70	20.10	20.40
2016-11-18	64.00	64.80	64.60	64.00	64.10	64.30	64.90	64.30	64.90	64.00	64.30	64.80	64.00	64.40	64.50	64.60	65.00	64.20	64.20	64.30	64.30	64.80	64.00	64.00
2016-11-19	2.00	2.60	2.40	3.00	2.20	2.70	2.10	2.90	2.40	2.10	2.60	2.30	2.90	2.30	2.10	2.60	2.40	2.20	2.10	2.50	2.30	2.40	2.80	2.00
2016-11-20	31.00	31.90	31.10	32.00	31.50	31.50	31.60	31.10	31.30	31.60	31.90	31.20	31.10	31.20	31.00	32.00	31.80	31.00	31.90	31.70	31.80	31.00	31.60	31.30
2016-11-21	18.00	18.40	18.50	18.20	18.50	18.00	18.20	18.00	18.60	18.50	18.30	18.90	18.90	18.10	18.90	18.70	19.00	18.50	19.00	18.00	19.00	18.90	18.50	18.80
2016-11-22	92.00	92.90	93.00	92.00	92.20	92.60	92.20	92.70	92.50	92.30	92.60	92.10	93.00	92.00	92.10	92.00	92.30	92.00	92.30	92.40	92.80	92.00	93.00	93.00
2016-11-23	6.00	6.10	7.00	6.40	7.00	6.20	6.30	6.60	6.10	6.50	6.30	6.40	6.50	7.00	6.90	6.20	6.80	6.90	6.50	6.30	6.10	6.10	6.00	6.10

Slika 4.7. Tablični prikaz podataka za parametar Vlažnost tla 1 [Izvor: autor]

Na slici 4.7. je prikazan tablični prikaz podataka za parametar Vlažnost tla 1. Klikom na jednu od stanica povlače se podaci iz baze za kliknutu stanicu i odabrani parametar. Podaci u tablici predstavljaju mjerenja zadnjih 7 dana. Tablica se sastoji od 24 stupca gdje svaki stupac predstavlja jedno mjerenje u danu.

4.3.1. Opis realizacije

U ovom dijelu rada će biti opisan proces kodiranja te će biti komentirani dijelovi koda. Nakon kreiranja i popunjavanja baze podataka slijedio je proces kodiranja. Prvo je osmišljena struktura direktorija i datoteka, a zatim i kreirana. Preuzete su sve potrebne datoteke programskih okvira te su smještene u planirani direktorij. Ispod se nalazi primjer koda koji predstavlja zaglavlje kreirane *index.html* datoteke.

Kod 4.3. Zaglavlje index.html datoteke

```
<head>
<title>Diplomski rad - Tomislav Šapina</title>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" type="text/css" href=" lib/leaflet/leaflet.css">
<link rel="stylesheet" type="text/css" href=" lib/bootstrap/css/bootstrap-
theme.min.css">
<link rel="stylesheet" type="text/css" href="
lib/bootstrap/css/bootstrap.min.css">
<link rel="stylesheet" type="text/css" href=" css/style.css">
<link rel="stylesheet" type="text/css" href="
lib/daterangepicker/daterangepicker.css">
<script type="text/javascript" src="lib/leaflet/leaflet.js"></script>
<script type="text/javascript" src="lib/bootstrap/js/npm.js"></script>
<script type="text/javascript" src="lib/jquery/jquery-3.1.1.min.js"></script>
<script type="text/javascript" src="data/hr-all.geo.json"></script>
<script type="text/javascript" src="lib/daterangepicker/moment.js"></script>
<script type="text/javascript"
src="lib/daterangepicker/daterangepicker.js"></script>
<script type="text/javascript" src="lib/bootstrap/js/bootstrap.js"></script>
<script type="text/javascript" src="lib/dygraph/dygraph-
combined.js"></script>
<script type="text/javascript"
src="lib/daterangepicker/waitingfor.js"></script>
<script type="text/javascript"
src="lib/daterangepicker/dateformat.js"></script>
</head>
```

Kod 4.3. predstavlja HTML kod koji definira zaglavlje *index.html* dokumenta. Sastoji se od oznake *meta* koja sadrži informacije potrebne za ispravno prikazivanje stranice. Slijedi oznaka *title* kojom se definira naziv stranice, zatim slijedi oznaka *link* koje sadrže poveznice na CSS dokumente. Uključeni CSS dokumenti predstavljaju CSS dokumente programskih okvira te dokument *style.css* u kojem je pisan dodatni CSS kod. U zaglavlju se još nalaze *script* oznake koje sadrže poveznicu na JavaScript dokumente. Svi JavaScript dokumenti definirani u zaglavlju su dokumenti koji se odnose na neki od programskih okvira. Osim JavaScript dokumenata u zaglavlju pod oznakom *script* nalazi se i poveznica na GeoJSON datoteku. JavaScript datoteke kojima je definirana logika aplikacije smještene su u *body* dijelu HTML dokumenta.

Kod 4.4. Uključivanje JavaScript datoteka

```
<script type="text/javascript" src="js/firstGraphicDisplay.js"></script>
<script type="text/javascript" src="js/secondGraphicDisplay.js"></script>
```

Kod 4.4. se odnosi na uključivanje spomenutih JavaScript datoteka u *body* dio HTML dokumenta. U datoteci *firstGraphicDisplay.js* je smješten kod pisan za grafički prikaz podataka na karti Hrvatske pa će u ovom djelu rada biti ukratko opisane funkcije koje sadrži ta datoteka. Spomenuta datoteka sadrži šest JavaScript funkcija, a to su: *createMap(), style(), dataForPreviousDay(), setLegend(), onClick()* i *detectParametarChange()*. Funkcije se pozivaju u *body* dijelu HTML dokumenta odmah ispod uključenih JavaScript datoteka.

Kod 4.5. Pozivanje JavaScript funkcija

```
<script type="text/javascript">
dataForPreviousDay();
createMap();
setLegend();
detectParametarChange();
</script>
```

Kod 4.5. prikazuje pozivanje nekih od prethodno navedenih JavaScript funkcija. Prva funkcija koja se poziva je *dataForPreviousDay()*. U njoj je smješten kod koji šalje AJAX zahtjev prema poslužitelju te radi operacije s dohvaćenim podacima. Sljedeća funkcija koja se poziva je *createMap()*. U njoj se kreira karta pomoću GeoJSON podataka, dodaju se stilovi na kartu te se postavljaju markeri na kartu. Funkcija *setlegend()* postavlja legendu na kartu ovisno o odabranom parametru na padajućem izborniku. Zadnja funkcija koja se poziva u *body* dijelu *index.html* dokumenta je *detectParametarChange()*. Njena funkcija je da se nakon svake promjene parametra u padajućem izborniku podaci na karti ažuriraju

Kod 4.6. Varijable u datoteci „firstGraphicDisplay.js“

```
var averages = [];
var colorBuffer = [];
var par = $("#map-par").val();
var map = L.map('map');
var position1;
var position2;
var position3;
var position4;
var croatiaLayer;
```

U kodu 4.6 su definirane varijable koje su korištene u navedenim funkcijama a kod je smješten na samom početku datoteke *firstGraphicDisplay.js* te su ispod njega pisane funkcije. Prva varijabla *averages* je polje koje služi kao spremnik za izračun prosječne vrijednosti mjerenja prethodnog dana za svaku stanicu. Izračun se vrši u funkciji *dataForPreviousDay()* nakon što su preuzeti podaci za prethodni dan. U polje *averages* se spremaju 4 vrijednosti, a svaka vrijednost predstavlja prosječnu vrijednost mjerenja nekog parametra za svaku stanicu. Popunjeno polje *averages* se koristi u funkciji *style()* pri bojanju karte, a nijansa boje karte za svaku stanicu ovisi o prosječnoj vrijednosti. Varijabla *colorBuffer* služi kao spremnik za boje. Boje se spremaju u funkciji *style()* a nijanse boje koje će biti spremljene ovisi o odabranom parametru. Služi za bojanje karte, a koristi se još u funkciji *setLegend()* pri kreiranju legende na karti. Vrijednosti varijable *par* ovisi o vrijednosti parametra na padajućem izborniku. Varijabli *map* je dodijeljen inicijalizirani objekt koji predstavlja kartu. Karta je kreirana u *div* elementu koji sadrži *id* naziva *map*. Varijablama *position1*, *position2*, *position3* i *position4* će biti dodijeljene vrijednosit koje predstavljaju poziciju markera postavljenih na karti. Varijabla *croatiaLayer* predstavlja GeoJSON objekt koji se dodaje prethodno kreiranom objektu *map*.

Kod 4.7. Prikaz funkcije *createMap()*

```
function createMap()
{
	//zoom disable
map.touchZoom.disable();
	map.doubleClickZoom.disable();
	map.scrollWheelZoom.disable();
	map.boxZoom.disable();
	map.keyboard.disable();
	$(".leaflet-control-zoom").css("visibility",
"hidden");

	croatiaLayer = L.geoJson(croatia, {style:
style}).addTo(map);
	map.fitBounds(croatiaLayer.getBounds());

	var point1 = [45.45667, 18.01667 ];
	var point2 = [45.36667, 16.51667 ];
	var point3 = [46.19667, 16.51667 ];
	var point4 = [45.16667, 15.11667 ];
```

```
            position1   =   L.marker(point1).addTo(map).on('click',
        onClick);
            position2   =   L.marker(point2).addTo(map).on('click',
        onClick);
            position3   =   L.marker(point3).addTo(map).on('click',
        onClick);
            position4   =   L.marker(point4).addTo(map).on('click',
        onClick);

        }
```

Kod 4.7. se odnosi na funkciju *createMap()* koja kreira kartu uz pomoć objekta *map*. Na početku koda su pozvane metode koje onemogućuju korisniku zumiranje na karti pomoću klika miša i tipkovnice. Funkcijom *$(".leaflet-control-zoom").css("visibility", "hidden");* je skriven HTML element koji omogućava zumiranje karte. Nakon toga je GeoJSON sloj dodan objektu *map* koji prikazuje kartu Hrvatske. Metoda GeoJson prima za parametar objekt *croatia* koji predstavlja GeoJSON podatke iz datoteke *hr-all.geo.json*. Također, prima za parametar funkciju *style()* koja dodaje odgovarajuću boju pojedinom području na karti. Navedene su četiri varijable *point1*, *point2*, *point3* te *point4* koje predstavljaju koordinate na kojima trebaju biti smješteni markeri na karti. Prvi marker je dodan linijom koda *L.marker(point1).addTo(map).on('click', onClick);* a ostali markeri su dodani na isti način. Metoda *marker()* prima kao parametar varijablu *point1*. Također je definirano da se klikom na marker poziva metoda *onClick()*.

Kod 4.8. Prikaz funkcije *dataForPreviousDay()*

```
function dataForPreviousDay()
{
      var station1Avg = 0;
      var station2Avg = 0;
      var station3Avg = 0;
      var station4Avg = 0;
      console.log("test");
      $.ajax({
            type: "POST",
            url: "php/getDataByPreviousDay.php",
            dataType: "json",
            async: false,
            data: {par: par },
            success: function(data)
```

```
        {
                console.log(data);
                for(i=0; i<data.length; i++)
                {
                        if(i<24)
                        {
                        station1Avg = station1Avg + parseFloat(data[i].par);
                        }
                        else if(i>=24 && i<48)
                        {
                        station2Avg += parseFloat(data[i].par);
                        }
                        else if(i>=48 && i<72)
                        {
                        station3Avg += parseFloat(data[i].par);
                        }
                        else
                        {
                        station4Avg += parseFloat(data[i].par);
                        }
                }

                averages.push(station1Avg/24);
                averages.push(station2Avg/24);
                averages.push(station3Avg/24);
                averages.push(station4Avg/24);

        }//end of success
});//end of ajax
}
```

Kod 4.8. predstavlja funkciju *dataForPreviousDay()*. Na početku funkcije kreirane su 4 varijable *station1Avg*, *station2Avg*, *station3Avg* te *station4Avg* u koje se spremaju prosječna vrijednost mjerenja za određenu stanicu i odabrani parametar. Nakon toga, definiran je AJAX zahtjev prema poslužitelju *POST* metodom na url *php/getDataByPreviousDay.php*. Prosljeđuje se varijabla *par* koja predstavlja odabrani parametar sa padajućeg izbornika. PHP skripta vraća kao rezultat uređene JSON podatke. Kroz JSON podatke se prolazi *for* petljom te se računaju prosječne vrijednosti mjerenja za svaku stanicu. Na kraju se sve varijable s izračunatim vrijednostima dodaju u polje *averages*.

Kod 4.9. Prikaz koda iz datoteke *getDataByPreviousDay.php*

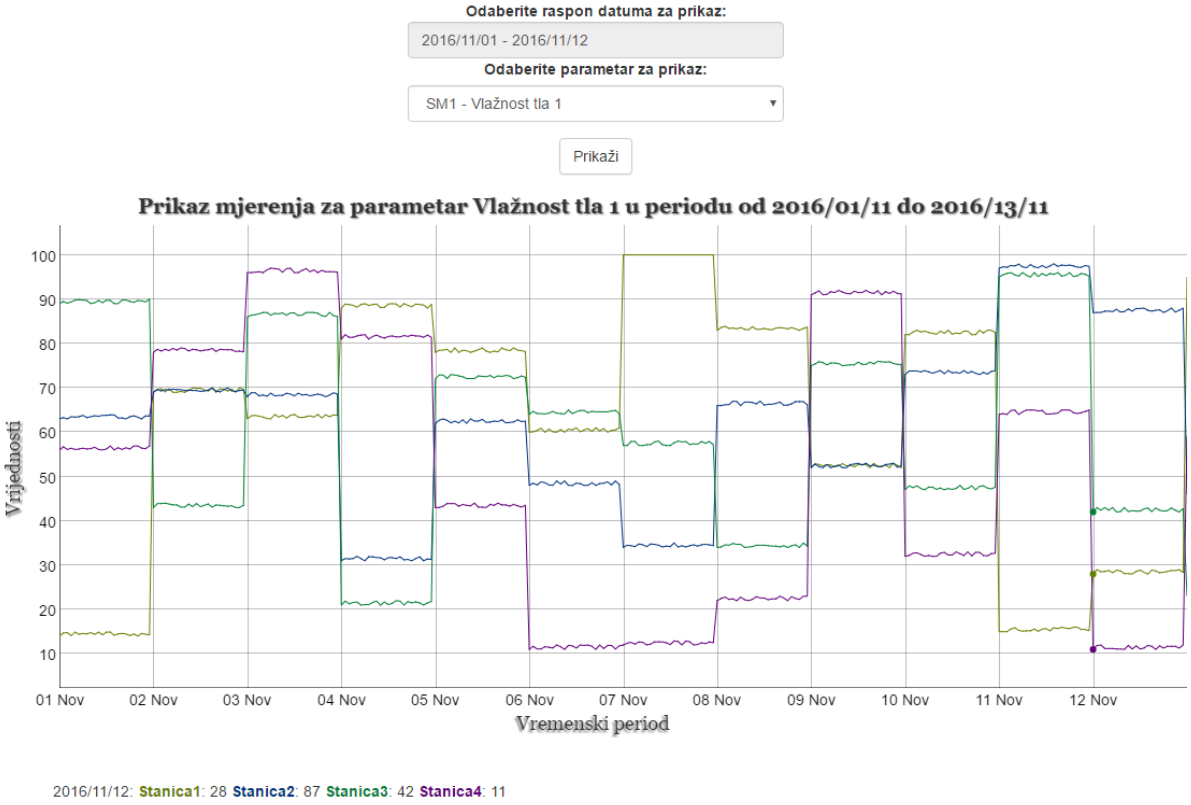
```
<?php
require 'connect.php';
$par = $_POST['par'];

if ($result =  $mysqli->query("SELECT stanica_id, datum, {$par} FROM podaci
WHERE datum between subdate(CURDATE(), 1) and CURDATE() ORDER BY stanica_id"))
{
	$numRows = $result->num_rows;
	if($numRows>0)
	{
		$counter = 0;
		//loop through selected data
		while($row = $result->fetch_row())
		{
			$respond[$counter]["stanica"] = $row[0];
			$respond[$counter]["datum"] = $row[1];
			$respond[$counter]["par"] = $row[2];
			$counter++;
		}
	}
	else
	{
		$respond['numRowsError'] = 0;
	}
}
else
{
	die("QUERY ERROR: " . $result->error);
}

echo json_encode($respond);
```

Kod 4.9. je smješten u PHP skripti *getDataByPreviousDay.php*. Na početku koda je uključena datoteka *connect.php* u kojoj se kreira objekt *$mysqli* koji predstavlja povezivanje na MySQL poslužitelj. Ispod je kreirana varijabla *$par* koja preuzima parametar iz super globalne varijable *$_POST* prosljeđen AJAX zahtjevom. Nakon toga se radi upit na bazu pomoću metode *query*() koja prima kao parametar SQL kod. Ako je upit uspješno izvršen kreira se objekt *$result* te ukoliko postoje podaci za prethodni dan onda se prolazi kroz dobivene podatke pomoću metode *feth_row()*

koristeći *while* petlju. Dohvaćeni podaci se spremaju u varijablu *$respond* koja se ispisuje u obliku JSON koda koji je ujedno i povratni rezultat na AJAX zahtjev.

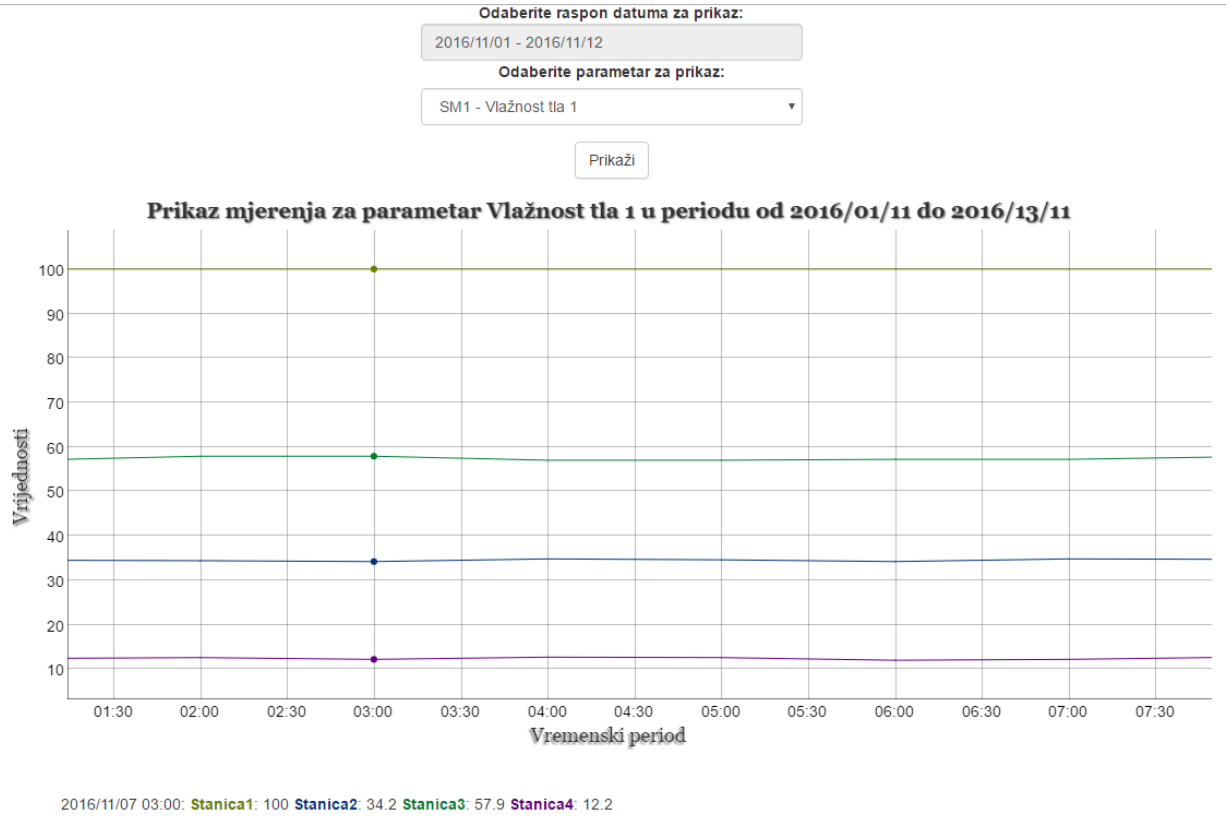
4.4. Prikaz podataka na linijskom grafu

U ovom dijelu rada će prvo biti prikazan rezultat izrade dijela aplikacije koji predstavlja grafički prikaz podataka na linijskom grafu te će biti opisani dijelovi programskog koda.



Slika 4.8. Grafički prikaz podataka na linijskom grafu [Izvor: autor]

Slika 4.8. prikazuje podatke na linijskom grafu. Omogućen je odabir raspona podataka te odabir parametra za prikaz. Klikom na gumb „prikaži“ iscrtava se linijski graf. Na x-osi je smješten vremenski period odabranog raspona datuma. Na y-osi je smješten raspon vrijednosti parametra. Na grafu prikazuju linije u različitim bojama gdje se svaka boja odnosi na jednu stanicu. Prelaskom miša preko linija ispod grafa se pokazuje datum te vrijednosti za taj datum za svaku stanicu. Dvoklikom je moguće zumirati graf kao što je prikazano na slici 4.9..



Slika 4.9. Grafički prikaz podataka na linijskom grafu [Izvor: autor]

Na slici 4.9. je prikazan zumirani pogled na linijski graf te je vidljivo da je omogućen pogled po satima. Na slijedećih nekoliko stranica će biti navedeni i opisani dijelovi kodova linijskog grafa.

Kod 4.10. Prikaz funkcije iz datoteke *secondGraphicDisplay.js*

```
$('#btn').click(function(){
            par = $("#parametar").val();
            var enteredDate= $("#reportrange").val(); // uzimanje vrijednosti
            //provjera dali postoji vrijednost,ako ne izbaci poruku
            if(!enteredDate) {
             $('#validationAlert').modal('show');
            }
            else
            {
              //split entered  date to startDate and endDate
              var splitedDateContainer = enteredDate.split('-');
              //get splited and trimed date
              var startDate = $.trim(splitedDateContainer[0]);
              var endDate = $.trim(splitedDateContainer[1]);
```

```
            $.ajax({
              type: "POST",
              url: "php/getDataByPar.php",
              dataType: "json",
              data: {startDate: startDate, endDate: endDate, parametar: par},
              success: function(data) {
          console.log(data);
                //provjerava data ima za property numRowsError
                if(data.hasOwnProperty('numRowsError')){
                   console.log("num rows error");
                  $('#noDataAlert').modal('show');
                }else{
                  secondCounter++;
                  waitingDialog.show('Graf se učitava...', {dialogSize: 'sm',
progressType: 'warning'});
                  setTimeout(function () {
                    waitingDialog.hide();
                    showGraph(data, secondCounter);
                  }, 1500);
                }

              },//end of success
              error: function (request, status, error) {
          console.log(request.responseText);
              }
            });//end of ajax

          }//end of if(!enteredDate)

        });//end of $('#save').click(function(){

      });//end of $(document).ready(function()
```

Kod 4.10. nalazi se u datoteci *secondGraphicDisplay.js* a izvodi se nakon što korisnik pritisne gumb „prikaži“. Prvo se uzimaju vrijednosti odabranog parametra, vrijednosti početnog i krajnjeg datuma te se spremaju u varijable. Nakon toga je definiran AJAX zahtjev prema poslužitelju putem *POST* metode na url *php/getDataByPar.php* . Prosljeđuju se varijabla *par*, *startDate* i *endDate*. PHP skripta vraća kao rezultat uređene JSON podatke. Nakon toga se podaci prosljeđuju u funkciju *showGraph()* koja je zadužena za prikazivanje grafa.

Kod 4.11. Kreiranje grafa

```
g2 = new Dygraph( document.getElementById("graphdiv2"), buffer,
                  {
                  labels:  [  "date",  "Stanica1",  "Stanica2",  "Stanica3",
            "Stanica4"],
                  legend: 'always',
                  title: 'Prikaz mjerenja za parametar ' + choosenPar + ' u
                  periodu od ' + startDate + ' do ' + endDate,
                  titleHeight: 32,
                  ylabel: 'Vrijednosti',
                  xlabel: 'Vremenski period',
                  }
                  );
```

Kod 4.11. se nalazi u datoteci *secondGraphicDisplay.js* te je zadužen za kreiranje grafa. Kreira se nova instanca objekta *Dygraph* pod nazivom *g2*. Prvi proslijeđeni parametar je *div* element naziva *graphdiv2* u kojem se prikazuje graf. Drugi parametar je polje *buffer* koji sadrži preuzete i uređene podatke za prikaz. Treći parametar je objekt koji sadrži podakte o naslovu grafa, o informacijama na grafu te x-osi i y-osi grafa.

5.ZAKLJUČAK

Zadatak ovog diplomskog rada bio je izraditi aplikaciju za grafički prikaz podataka prikupljenih uporabom senzorske mreže. Podaci predstavljaju parametre za meteo stanicu. U nedostatku originalnih podataka s meteo stanica generirani su podaci koji predstavljaju očekivane vrijednosti na pojedinim geografskim lokacijama i služe isključivo u demonstracijske svrhe za prikaz mogučnosti obrađivanih tehnologija. Korisniku je omogućen prikaz podataka na karti Hrvatske te na linijskom grafu. Prvi dio rada se odnosio na istraživanje tehnologija i odabir alata koji olakšavaju izradu aplikacije. Sve odabrane tehnologije, programski okviri i alati su besplatni te kao takvi nisu ograničeni u tehničkom smislu. Također su dobro dokumentirani i podržani od programerske zajednice. Drugi dio se odnosio na kreiranje baze sa podacima i izradu koda aplikacije. Valja napomenuti kako programski okviri koji se koriste u aplikaciji višestruko ubrzavaju proces izrade aplikacije. Treći dio rada se odnosio na pisanje samog rada koji detaljno opisuje tehnologije i proces izrade aplikacije. Web aplikacija je u ovoj fazi razvoja jednostavna te ju je moguće unaprijediti na nekoliko načina. Prvenstveno, moguće je dodati više različitih grafičkih prikaza podataka. Nadalje, može se omogućiti izvoz podataka u .xls i .pdf format. Zatim je moguće implementirati API uslugu koja pruža originalne podatke. Nakon izrade aplikacije na lokalnom poslužitelju, aplikacija je premještena na adresu http://www.etfos.unios.hr/~tsapina/.

LITERATURA

[1] HTML Tutorial, URL: http://www.w3schools.com/html/, studeni 2016.

[2] CSS Tutorial, URL: http://www.w3schools.com/css/, studeni 2016.

[3] D. Crockford, JavaScript: The Good Parts, O'Reilly Media , 2008.

[4] AJAX Tutorial, URL: http://www.w3schools.com/xml/ajax_intro.asp/, studeni 2016.

[5] K. Tatroe, Programming PHP, 3rd Edition, O'Reilly Media , 2013.

[6] Bootstrap Tutorial, URL: http://getbootstrap.com/, studeni 2016.

[7] jQuery Tutorial, URL: http://www.w3schools.com/jquery/, studeni 2016.

[8] Date range picker Tutorial, URL: http://www.daterangepicker.com/, studeni 2016.

[9] JSON Tutorial, URL: http://www.w3schools.com/js/js_json_intro.asp/ , studeni 2016.

[10] GeoJSON Tutorial, URL: http://geojson.org/, studeni 2016.

[11] Leaflet Tutorial, URL: http://leafletjs.com/, studeni 2016.

[12] DyGraphs Tutorial, URL: http://dygraphs.com/, studeni 2016.

[13] XAMPP Tutorial, URL: https://www.apachefriends.org/index.html, studeni 2016.

[14] phpMyAdmin Tutorial, URL: https://www.phpmyadmin.net/, studeni 2016.

SAŽETAK

Diplomski rad podijeljen je na praktični i teorijski dio. Praktični dio se odnosi na izradu web aplikacije za grafički prikaz podataka prikupljenih uporabom senzorske mreže. Podaci predstavljaju parametre za meteo stanicu te su prikazani na karti Hrvatske i na linijskom grafu. U svrhu izrade ovog rada podaci će se generirati prema očekivanim mogućim vrijednostima za odabrene parametre. Tehnologije korištene pri izradi aplikacije su HTML, CSS, JavaScript, PHP i SQL. Za grafički prikaz podataka korišteni su programski okviri Leaflet i DyGraphs. U teorijskom dijelu su opisane tehnologije i alati koji su korišteni pri izradi aplikacije. Opisan je proces izrade te su prikazani rezultati izrade.

Ključne riječi: HTML, CSS, JavaScript, PHP, SQL, Leaflet, DyGraphs

ABSTRACT

This thesis is divided into theoretical and practical parts. The practical part describes the process of building a web application for meteoorological data visualization. The practical part is related to the development of web applications for data visualization of data, collected using a sensor grid. Data represents the parameters for the meteo station and is shown on a geographical map of Croatia and on a line graph. For the purpose of this thesis, research data will be generated according to the expected possible values for choosen parameters. The technologies used in the preparation of applications are HTML, CSS, JavaScript, PHP and SQL. For the purpose of data visualization programming frameworks Leaflet and DyGraphs were used. The theoretical part describes the technologies,tools and the build process that were used in the preparation of applications. Also, the theoretical part contains the final result and analysis.

Key words: HTML, CSS, JavaScript, PHP, SQL, Leaflet, DyGraphs

ŽIVOTOPIS

Šapina Tomislav rođen je 17.ožujka 1991. godine u Gradačcu u Bosni i Hercegovini. Živi u ulici Ivana Meštrovića 28, u Piškorevcima (Đakovo). Osnovnu školu „Matija Gubec“ pohađao je u Piškorevcima od 1997. do 2005. godine. Srednju strukovnu školu Braće Radića, smjer Ekonomist pohađao je u Đakovu od 2005. do 2010. godine. Završetkom srednje škole upisuje Elektrotehnički fakultet Osijek, stručni studij elektrotehnike, smjer Informatika u listopadu 2010. godine koji redovito završava. Zatim upisuje razlikovnu godinu na Elektrotehničkom fakultetu te ju završava, svoj studij nastavlja na Diplomskom studiju, smjer Procesno računalstvo.

Na ETFOS-u stječe znanja iz web programiranja (HTML, CSS, JavaScript, PHP), baza podataka (SQL), objektno-orijentiranog programiranja (C++, C#, JAVA). Također usavršava vladanje alatima Microsoft Office-a (Word, Excel, PowerPoint).

.

PRILOG A. Programski kod aplikacije

Index.html

```
<!DOCTYPE html>
<html>
<head>

	<title>Diplomski rad - Tomislav Šapina</title>
	<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">

	<link rel="stylesheet" type="text/css" href=" lib/leaflet/leaflet.css">
	<link rel="stylesheet" type="text/css" href=" lib/bootstrap/css/bootstrap-theme.min.css">
	<link rel="stylesheet" type="text/css" href=" lib/bootstrap/css/bootstrap.min.css">
	<link rel="stylesheet" type="text/css" href=" css/style.css">
	<link rel="stylesheet" type="text/css" href=" lib/daterangepicker/daterangepicker.css">

	<script type="text/javascript" src="lib/leaflet/leaflet.js"></script>
	<script type="text/javascript" src="lib/bootstrap/js/npm.js"></script>
	<script type="text/javascript" src="lib/jquery/jquery-3.1.1.min.js"></script>
	<script type="text/javascript" src="data/hr-all.geo.json"></script>
	<script type="text/javascript" src="lib/daterangepicker/moment.js"></script>
	<script type="text/javascript" src="lib/daterangepicker/daterangepicker.js"></script>
	<script type="text/javascript" src="lib/bootstrap/js/bootstrap.js"></script>
	<script type="text/javascript" src="lib/dygraph/dygraph-combined.js"></script>
	<script type="text/javascript" src="lib/daterangepicker/waitingfor.js"></script>
	<script type="text/javascript" src="lib/daterangepicker/dateformat.js"></script>

</head>

<body>
<div class="container">
	<div class="app-wrapper">
		<div class="jumbotron">
	<h3>Grafički prikaz podataka prikupljenih uporabom senzorske mreže</h3>
	</div>

	<form class="form-horizontal">
			<div class="form-group">
			<div class="col-md-4 col-md-offset-4">
```

```
                        <label >Odaberite parametar za prikaz:</label>
                                <select class="form-control" id="map-par">
                                <option value="sm1">SM1 - Vlažnost tla 1</option>
                                <option value="sm2">SM2 - Vlažnost tla 2</option>
                                <option value="sm3">SM3 - Vlažnost tla 3</option>
                                <option value="sm4">SM4 - Vlažnost tla 4</option>
                                <option value="sm5">SM5 - Vlažnost tla 5</option>s
                                <option value="sm6">SM6 - Vlažnost tla 6</option>
                                <option value="ah1">AH1 - Vlažnost zraka</option>
                                <option value="at1">AT1 - Temperatura zraka</option>
                                <option value="ws1">WS1 - Brzina vjetra</option>
                                <option value="pp1">PP1 - Padaline</option>
                                </select>
                        </div><!-- /.col-md-4 col-md-offset-4 -->
                        </div><!-- form-group-->
                </form>
                        <div class="station-desc">
                <hr>
                <h2>Popis stanica:</h2>
                <p>
                <b>Stanica 1:</b> Osječko-Baranjska, Vukovarsko-Srijemska, Požesko-Slavonska,
Brodsko-Posavska, Virovitičko-Podravska <br>
                        <b>Stanica 2:</b> Sisačko-Moslavačka, Bjelovarska-Bilogorska, Zagrebačka,
Grad Zagreb <br>
                        <b>Stanica 3:</b> Međimurska, Varaždinska, Krapinsko-Zagorska, Koprivnicko-
Križevacka <br>
                        <b>Stanica 4:</b> Sisačko-Moslavačka, Bjelovarska-Bilogorska, Zagrebačka,
Grad Zagreb <br>
                        </hr>
                        </p>
                </div>
                <div id="map"></div>
                <strong>Odaberite raspon datuma za prikaz:</strong>
                <form class="form-horizontal">
                        <div class="form-group">
                        <div class="col-md-4 col-md-offset-4">
                                <input type="text" id="reportrange" class="form-control"
readonly="readonly" value="" />
                                <input type="hidden" name="to" id="to" value="">
                                <input type="hidden" name="from" id="from" value="">

                                <label >Odaberite parametar za prikaz:</label>
                                        <select class="form-control" id="parametar">
```

```
                                    <option value="sm1">SM1 - Vlažnost tla 1</option>
                                    <option value="sm2">SM2 - Vlažnost tla 2</option>
                                    <option value="sm3">SM3 - Vlažnost tla 3</option>
                                    <option value="sm4">SM4 - Vlažnost tla 4</option>
                                    <option value="sm5">SM5 - Vlažnost tla 5</option>s
                                    <option value="sm6">SM6 - Vlažnost tla 6</option>
                                    <option value="ah1">AH1 - Vlažnost zraka</option>
                                    <option value="at1">AT1 - Temperatura zraka</option>
                                    <option value="ws1">WS1 - Brzina vjetra</option>
                                    <option value="pp1">PP1 - Padaline</option>
                                    </select>
                        </div><!-- /.col-md-4 col-md-offset-4 -->
                        </div><!-- form-group-->
                        <div class="form-group">
                        <button type='button' class="btn btn-default" id="save">Prikaži</button>
                        </div><!-- form-group-->
                </form>
<div id="graphdiv2" style="max-width:100%; height:500px;"></div>
</div><!-- /.app-wrapper -->
</div><!-- /.container -->

<script type="text/javascript" src="js/firstGraphicDisplay.js"></script>
<script type="text/javascript" src="js/secondGraphicDisplay.js"></script>

<script type="text/javascript">
        dataForPreviousDay();
        createMap();
        setLegend();
        detectParametarChange();
</script>

<div class="modal fade" tabindex="-1" role="dialog" id="validationAlert">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-
hidden="true">&times;</span></button>
</div>
<div class="modal-body">
<p>Unesite datume!!</p>
</div>
```

```
<div class="modal-footer">
<button type="button" class="btn btn-default" data-dismiss="modal">U redu</button>
</div>
</div><!-- /.modal-content -->
</div><!-- /.modal-dialog -->
</div><!-- /.modal -->

<div class="modal fade" tabindex="-1" role="dialog" id="noDataAlert">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-
hidden="true">&times;</span></button>
</div>
<div class="modal-body">
<p>U unesenom rangu nema podataka za prikaz!</p>
</div>
<div class="modal-footer">
<button type="button" class="btn btn-default" data-dismiss="modal">U redu</button>
</div>
</div><!-- /.modal-content -->
</div><!-- /.modal-dialog -->
</div><!-- /.modal -->

</body>
</html>
```

firstGraphicDisplay.js

```
var averages = [];
var colorBuffer = [];
var par = $("#map-par").val();
var map = L.map('map');
var info = L.control();
var position1;
var position2;
var position3;
var position4;
var croatiaLayer;

console.log("par" + par);
function style(feature)
{
	var averageStation1 = averages[0];
	var averageStation2 = averages[1];
	var averageStation3 = averages[2];
	var averageStation4 = averages[3];

	var colorStation1;
	var colorStation2;
	var colorStation3;
	var colorStation4;
	var colorRest = "#ffffff";

	if(par == "sm1" || par == "sm2" || par == "sm3" || par == "sm4" || par == "sm5" || par == "sm6")
	{
		colorBuffer = [];
		colorBuffer.push("#ffd1b3");
		colorBuffer.push("#ffc299");
		colorBuffer.push("#ffb380");
		colorBuffer.push("#ff944d");
		colorBuffer.push("#ff8533");
		colorBuffer.push("#e65c00");
		colorBuffer.push("#cc5200");
		colorBuffer.push("#b34700");
		colorBuffer.push("#803300");
		colorBuffer.push("#662900");
```

```
                            if(averageStation1<=10){ colorStation1 = colorBuffer[0];}
                            else if(averageStation1>10 && averageStation1 <= 20) {colorStation1
= colorBuffer[1]; }
                            else if(averageStation1>20 && averageStation1 <= 30) {colorStation1
= colorBuffer[2]; }
                            else if(averageStation1>30 && averageStation1 <= 40) {colorStation1
= colorBuffer[3]; }
                            else if(averageStation1>40 && averageStation1 <= 50) {colorStation1
= colorBuffer[4]; }
                            else if(averageStation1>50 && averageStation1 <= 60) {colorStation1
= colorBuffer[5]; }
                            else if(averageStation1>60 && averageStation1 <= 70) {colorStation1
= colorBuffer[6]; }
                            else if(averageStation1>70 && averageStation1 <= 80) {colorStation1
= colorBuffer[7]; }
                            else if(averageStation1>80 && averageStation1 <= 90) {colorStation1
= colorBuffer[8]; }
                            else if(averageStation1>90 && averageStation1 <= 110)
{colorStation1 = colorBuffer[9]; }


                            if(averageStation2<=10){ colorStation2 = colorBuffer[0];}
                            else if(averageStation2>10 && averageStation2 <= 20) {colorStation2
= colorBuffer[1]; }
                            else if(averageStation2>20 && averageStation2 <= 30) {colorStation2
= colorBuffer[2]; }
                            else if(averageStation2>30 && averageStation2 <= 40) {colorStation2
= colorBuffer[3]; }
                            else if(averageStation2>40 && averageStation2 <= 50) {colorStation2
= colorBuffer[4]; }
                            else if(averageStation2>50 && averageStation2 <= 60) {colorStation2
= colorBuffer[5]; }
                            else if(averageStation2>60 && averageStation2 <= 70) {colorStation2
= colorBuffer[6]; }
                            else if(averageStation2>70 && averageStation2 <= 80) {colorStation2
= colorBuffer[7]; }
                            else if(averageStation2>80 && averageStation2 <= 90) {colorStation2
= colorBuffer[8]; }
                            else if(averageStation2>90 && averageStation2 <= 110)
{colorStation2 = colorBuffer[9]; }


                            if(averageStation3<=10){ colorStation3 = colorBuffer[0];}
                            else if(averageStation3>10 && averageStation3 <= 20) {colorStation3
= colorBuffer[1]; }
                            else if(averageStation3>20 && averageStation3 <= 30) {colorStation3
= colorBuffer[2]; }
                            else if(averageStation3>30 && averageStation3 <= 40) {colorStation3
= colorBuffer[3]; }
                            else if(averageStation3>40 && averageStation3 <= 50) {colorStation3
= colorBuffer[4]; }
                            else if(averageStation3>50 && averageStation3 <= 60) {colorStation3
= colorBuffer[5]; }
```

```
                                else if(averageStation3>60 && averageStation3 <= 70) {colorStation3
= colorBuffer[6]; }
                                else if(averageStation3>70 && averageStation3 <= 80) {colorStation3
= colorBuffer[7]; }
                                else if(averageStation3>80 && averageStation3 <= 90) {colorStation3
= colorBuffer[8]; }
                                else if(averageStation3>90 && averageStation3 <= 110)
{colorStation3 = colorBuffer[9]; }


                                if(averageStation4<=10){ colorStation4 = colorBuffer[0];}
                                else if(averageStation4>10 && averageStation4 <= 20) {colorStation4
= colorBuffer[1]; }
                                else if(averageStation4>20 && averageStation4 <= 30) {colorStation4
= colorBuffer[2]; }
                                else if(averageStation4>30 && averageStation4 <= 40) {colorStation4
= colorBuffer[3]; }
                                else if(averageStation4>40 && averageStation4 <= 50) {colorStation4
= colorBuffer[4]; }
                                else if(averageStation4>50 && averageStation4 <= 60) {colorStation4
= colorBuffer[5]; }
                                else if(averageStation4>60 && averageStation4 <= 70) {colorStation4
= colorBuffer[6]; }
                                else if(averageStation4>70 && averageStation4 <= 80) {colorStation4
= colorBuffer[7]; }
                                else if(averageStation4>80 && averageStation4 <= 90) {colorStation4
= colorBuffer[8]; }
                                else if(averageStation4>90 && averageStation4 <= 110)
{colorStation4 =colorBuffer[9]; }


                        }
                        else if(par == "ah1")
                        {
                                colorBuffer = [];
                                colorBuffer.push("#ccebff");
                                colorBuffer.push("#99d6ff");
                                colorBuffer.push("#80ccff");
                                colorBuffer.push("#66c2ff");
                                colorBuffer.push("#4db8ff");
                                colorBuffer.push("#33adff");
                                colorBuffer.push("#1aa3ff");
                                colorBuffer.push("#0099ff");
                                colorBuffer.push("#007acc");
                                colorBuffer.push("#006bb3");


                                if(averageStation1<=10){ colorStation1 = colorBuffer[0];}
                                else if(averageStation1>10 && averageStation1 <= 20) {colorStation1
= colorBuffer[1]; }
```

```
                            else if(averageStation1>20 && averageStation1 <= 30) {colorStation1
=  colorBuffer[2]; }

                            else if(averageStation1>30 && averageStation1 <= 40) {colorStation1
=  colorBuffer[3]; }

                            else if(averageStation1>40 && averageStation1 <= 50) {colorStation1
=  colorBuffer[4]; }

                            else if(averageStation1>50 && averageStation1 <= 60) {colorStation1
=  colorBuffer[5]; }

                            else if(averageStation1>60 && averageStation1 <= 70) {colorStation1
=  colorBuffer[6]; }

                            else if(averageStation1>70 && averageStation1 <= 80) {colorStation1
=  colorBuffer[7]; }

                            else if(averageStation1>80 && averageStation1 <= 90) {colorStation1
=  colorBuffer[8]; }

                            else if(averageStation1>90 && averageStation1 <= 110)
{colorStation1 = colorBuffer[9]; }


                            if(averageStation2<=10){ colorStation2 = colorBuffer[0];;}

                            else if(averageStation2>10 && averageStation2 <= 20) {colorStation2
=  colorBuffer[1];}

                            else if(averageStation2>20 && averageStation2 <= 30) {colorStation2
=  colorBuffer[2]; }

                            else if(averageStation2>30 && averageStation2 <= 40) {colorStation2
= colorBuffer[3]; }

                            else if(averageStation2>40 && averageStation2 <= 50) {colorStation2
=  colorBuffer[4]; }

                            else if(averageStation2>50 && averageStation2 <= 60) {colorStation2
=  colorBuffer[5]; }

                            else if(averageStation2>60 && averageStation2 <= 70) {colorStation2
=  colorBuffer[6]; }

                            else if(averageStation2>70 && averageStation2 <= 80) {colorStation2
=  colorBuffer[7]; }

                            else if(averageStation2>80 && averageStation2 <= 90) {colorStation2
=  colorBuffer[8]; }

                            else if(averageStation2>90 && averageStation2 <= 110)
{colorStation2 =  colorBuffer[9]; }


                            if(averageStation3<=10){ colorStation3 =  colorBuffer[0];}

                            else if(averageStation3>10 && averageStation3 <= 20) {colorStation3
=  colorBuffer[1]; }

                            else if(averageStation3>20 && averageStation3 <= 30) {colorStation3
=  colorBuffer[2]; }

                            else if(averageStation3>30 && averageStation3 <= 40) {colorStation3
=  colorBuffer[3]; }

                            else if(averageStation3>40 && averageStation3 <= 50) {colorStation3
=  colorBuffer[4]; }

                            else if(averageStation3>50 && averageStation3 <= 60) {colorStation3
=  colorBuffer[5]; }

                            else if(averageStation3>60 && averageStation3 <= 70) {colorStation3
=  colorBuffer[6]; }

                            else if(averageStation3>70 && averageStation3 <= 80) {colorStation3
=  colorBuffer[7]; }
```

```
                            else if(averageStation3>80 && averageStation3 <= 90) {colorStation3
=  colorBuffer[8]; }

                            else if(averageStation3>90 && averageStation3 <= 110)
{colorStation3 = colorBuffer[9]; }


                            if(averageStation4<=10){ colorStation4 =  colorBuffer[0];}

                            else if(averageStation4>10 && averageStation4 <= 20) {colorStation4
=  colorBuffer[1]; }

                            else if(averageStation4>20 && averageStation4 <= 30) {colorStation4
=  colorBuffer[2]; }

                            else if(averageStation4>30 && averageStation4 <= 40) {colorStation4
=  colorBuffer[3]; }

                            else if(averageStation4>40 && averageStation4 <= 50) {colorStation4
=  colorBuffer[4]; }

                            else if(averageStation4>50 && averageStation4 <= 60) {colorStation4
=  colorBuffer[5]; }

                            else if(averageStation4>60 && averageStation4 <= 70) {colorStation4
=  colorBuffer[6]; }

                            else if(averageStation4>70 && averageStation4 <= 80) {colorStation4
=  colorBuffer[7]; }

                            else if(averageStation4>80 && averageStation4 <= 90) {colorStation4
=  colorBuffer[8]; }

                            else if(averageStation4>90 && averageStation4 <= 110)
{colorStation4 =  colorBuffer[9]; }

                    }

                    else if(par == "at1")

                    {

                            colorBuffer = [];

                            colorBuffer.push("#ffe6e6");

                            colorBuffer.push("#ffcccc");

                            colorBuffer.push("#ffb3b3");

                            colorBuffer.push("#ff9999");

                            colorBuffer.push("#ff8080");

                            colorBuffer.push("#ff6666");



                            if(averageStation1>-25 && averageStation1 <= -10){ colorStation1 =
colorBuffer[0];}

                            else if(averageStation1>-10 && averageStation1 <= 0) {colorStation1
= colorBuffer[1]; }

                            else if(averageStation1>0 && averageStation1 <= 10) {colorStation1
=  colorBuffer[2]; }

                            else if(averageStation1>10 && averageStation1 <= 20) {colorStation1
=  colorBuffer[3]; }

                            else if(averageStation1>20 && averageStation1 <= 30) {colorStation1
=  colorBuffer[4]; }

                            else if(averageStation1>30 && averageStation1 <= 45) {colorStation1
=  colorBuffer[5]; }
```

```
                        if(averageStation2>-25 && averageStation2 <= -10){ colorStation2 =
colorBuffer[0];}
                        else if(averageStation2>-10 && averageStation2 <= 0) {colorStation2
= colorBuffer[1]; }
                        else if(averageStation2>0 && averageStation2 <= 10) {colorStation2
=  colorBuffer[2]; }
                        else if(averageStation2>10 && averageStation2 <= 20) {colorStation2
=  colorBuffer[3]; }
                        else if(averageStation2>20 && averageStation2 <= 30) {colorStation2
=  colorBuffer[4]; }
                        else if(averageStation2>30 && averageStation2 <= 45) {colorStation2
=  colorBuffer[5]; }


                        if(averageStation3>-25 && averageStation3 <= -10){ colorStation3 =
colorBuffer[0];}
                        else if(averageStation3>-10 && averageStation3 <= 0) {colorStation3
= colorBuffer[1]; }
                        else if(averageStation3>0 && averageStation3 <= 10) {colorStation3
=  colorBuffer[2]; }
                        else if(averageStation3>10 && averageStation3 <= 20) {colorStation3
=  colorBuffer[3]; }
                        else if(averageStation3>20 && averageStation3 <= 30) {colorStation3
=  colorBuffer[4]; }
                        else if(averageStation3>30 && averageStation3 <= 45) {colorStation3
=  colorBuffer[5]; }


                        if(averageStation4>-25 && averageStation4 <= -10){ colorStation4 =
colorBuffer[0];}
                        else if(averageStation4>-10 && averageStation4 <= 0) {colorStation4
= colorBuffer[1]; }
                        else if(averageStation4>0 && averageStation4 <= 10) {colorStation4
=  colorBuffer[2]; }
                        else if(averageStation4>10 && averageStation4 <= 20) {colorStation4
=  colorBuffer[3]; }
                        else if(averageStation4>20 && averageStation4 <= 30) {colorStation4
=  colorBuffer[4]; }
                        else if(averageStation4>30 && averageStation4 <= 45) {colorStation4
=  colorBuffer[5]; }
                }
                else if(par == "ws1")
                {
                        colorBuffer = [];
                        colorBuffer.push("#99ccff");
                        colorBuffer.push("#1a8cff");
                        colorBuffer.push("#0066cc");
                        colorBuffer.push("#003366");

                        if(averageStation1<=10){ colorStation1 = colorBuffer[0];}
                        else if(averageStation1>10 && averageStation1 <= 20) {colorStation1
= colorBuffer[1]; }
```

```
                        else if(averageStation1>20 && averageStation1 <= 30) {colorStation1
=  colorBuffer[2]; }
                        else if(averageStation1>30 && averageStation1 <= 45) {colorStation1
=  colorBuffer[3]; }


                        if(averageStation2<=10){ colorStation2 = colorBuffer[0];;}
                        else if(averageStation2>10 && averageStation2 <= 20) {colorStation2
=  colorBuffer[1];}
                        else if(averageStation2>20 && averageStation2 <= 30) {colorStation2
=  colorBuffer[2]; }
                        else if(averageStation2>30 && averageStation2 <= 45) {colorStation2
= colorBuffer[3]; }


                        if(averageStation3<=10){ colorStation3 =  colorBuffer[0];}
                        else if(averageStation3>10 && averageStation3 <= 20) {colorStation3
=  colorBuffer[1]; }
                        else if(averageStation3>20 && averageStation3 <= 30) {colorStation3
=  colorBuffer[2]; }
                        else if(averageStation3>30 && averageStation3 <= 45) {colorStation3
=  colorBuffer[3]; }


                        if(averageStation4<=10){ colorStation4 =  colorBuffer[0];}
                        else if(averageStation4>10 && averageStation4 <= 20) {colorStation4
=  colorBuffer[1]; }
                        else if(averageStation4>20 && averageStation4 <= 30) {colorStation4
=  colorBuffer[2]; }
                        else if(averageStation4>30 && averageStation4 <= 45) {colorStation4
=  colorBuffer[3]; }


                }
                else if(par == "pp1")
                {
                        colorBuffer = [];
                        colorBuffer.push("#d4d6d8");
                        colorBuffer.push("#e6f2ff");
                        colorBuffer.push("#cce6ff");
                        colorBuffer.push("#b3d9ff");
                        colorBuffer.push("#99ccff");
                        colorBuffer.push("#80bfff");
                        colorBuffer.push("#66b3ff");
                        colorBuffer.push("#4da6ff");
                        colorBuffer.push("#3399ff");
                        colorBuffer.push("#1a8cff");
                        colorBuffer.push("#0066cc");
                        colorBuffer.push("#004d99");
```

```
                            if(averageStation1==0){ colorStation1 = colorBuffer[0];}
                            else if(averageStation1>0 && averageStation1 <= 5) {colorStation1 =
colorBuffer[1]; }
                            else if(averageStation1>5 && averageStation1 <= 10) {colorStation1
=  colorBuffer[2]; }
                            else if(averageStation1>10 && averageStation1 <= 15) {colorStation1
=  colorBuffer[3]; }
                            else if(averageStation1>15 && averageStation1 <= 20) {colorStation1
=  colorBuffer[4]; }
                            else if(averageStation1>20 && averageStation1 <= 25) {colorStation1
=  colorBuffer[5]; }
                            else if(averageStation1>25 && averageStation1 <= 30) {colorStation1
=  colorBuffer[6]; }
                            else if(averageStation1>30 && averageStation1 <= 35) {colorStation1
=  colorBuffer[7]; }
                            else if(averageStation1>35 && averageStation1 <= 40) {colorStation1
=  colorBuffer[8]; }
                            else if(averageStation1>40 && averageStation1 <= 45) {colorStation1
= colorBuffer[9]; }
                            else if(averageStation1>45 && averageStation1 <= 50) {colorStation1
= colorBuffer[10]; }
                            else if(averageStation1>50 && averageStation1 <= 110)
{colorStation1 = colorBuffer[11]; }

                            if(averageStation2==0){ colorStation2 = colorBuffer[0];}
                            else if(averageStation2>0 && averageStation2 <= 5) {colorStation2 =
colorBuffer[1]; }
                            else if(averageStation2>5 && averageStation2 <= 10) {colorStation2
=  colorBuffer[2]; }
                            else if(averageStation2>10 && averageStation2 <= 15) {colorStation2
=  colorBuffer[3]; }
                            else if(averageStation2>15 && averageStation2 <= 20) {colorStation2
=  colorBuffer[4]; }
                            else if(averageStation2>20 && averageStation2 <= 25) {colorStation2
=  colorBuffer[5]; }
                            else if(averageStation2>25 && averageStation2 <= 30) {colorStation2
=  colorBuffer[6]; }
                            else if(averageStation2>30 && averageStation2 <= 35) {colorStation2
=  colorBuffer[7]; }
                            else if(averageStation2>35 && averageStation2 <= 40) {colorStation2
=  colorBuffer[8]; }
                            else if(averageStation2>40 && averageStation2 <= 45) {colorStation2
= colorBuffer[9]; }
                            else if(averageStation2>45 && averageStation2 <= 50) {colorStation2
= colorBuffer[10]; }
                            else if(averageStation2>50 && averageStation2 <= 110)
{colorStation2 = colorBuffer[11]; }

                            if(averageStation3==0){ colorStation3 = colorBuffer[0];}
                            else if(averageStation3>0 && averageStation3 <= 5) {colorStation3 =
colorBuffer[1]; }
```

```
                            else if(averageStation3>5 && averageStation3 <= 10) {colorStation3
=  colorBuffer[2]; }

                            else if(averageStation3>10 && averageStation3 <= 15) {colorStation3
=  colorBuffer[3]; }

                            else if(averageStation3>15 && averageStation3 <= 20) {colorStation3
=  colorBuffer[4]; }

                            else if(averageStation3>20 && averageStation3 <= 25) {colorStation3
=  colorBuffer[5]; }

                            else if(averageStation3>25 && averageStation3 <= 30) {colorStation3
=  colorBuffer[6]; }

                            else if(averageStation3>30 && averageStation3 <= 35) {colorStation3
=  colorBuffer[7]; }

                            else if(averageStation3>35 && averageStation3 <= 40) {colorStation3
=  colorBuffer[8]; }

                            else if(averageStation3>40 && averageStation3 <= 45) {colorStation3
= colorBuffer[9]; }

                            else if(averageStation3>45 && averageStation3 <= 50) {colorStation3
= colorBuffer[10]; }

                            else if(averageStation3>50 && averageStation3 <= 110)
{colorStation3 = colorBuffer[11]; }


                            if(averageStation4==0){ colorStation4 = colorBuffer[0];}

                            else if(averageStation4>0 && averageStation4 <= 5) {colorStation4 =
colorBuffer[1]; }

                            else if(averageStation4>5 && averageStation4 <= 10) {colorStation4
=  colorBuffer[2]; }

                            else if(averageStation4>10 && averageStation4 <= 15) {colorStation4
=  colorBuffer[3]; }

                            else if(averageStation4>15 && averageStation4 <= 20) {colorStation4
=  colorBuffer[4]; }

                            else if(averageStation4>20 && averageStation4 <= 25) {colorStation4
=  colorBuffer[5]; }

                            else if(averageStation4>25 && averageStation4 <= 30) {colorStation4
=  colorBuffer[6]; }

                            else if(averageStation4>30 && averageStation4 <= 35) {colorStation4
=  colorBuffer[7]; }

                            else if(averageStation4>35 && averageStation4 <= 40) {colorStation4
=  colorBuffer[8]; }

                            else if(averageStation4>40 && averageStation4 <= 45) {colorStation4
= colorBuffer[9]; }

                            else if(averageStation4>45 && averageStation4 <= 50) {colorStation4
= colorBuffer[10]; }

                            else if(averageStation4>50 && averageStation4 <= 110)
{colorStation4 = colorBuffer[11]; }

                     }


                      switch (feature.properties.name)

                      {

                            //Stanica1

                            case 'Osjecko-Baranjska':   return {color: colorStation1, opacity:
1, weight: 5, dashArray: '5',fillOpacity: 1};
```

```
                            case 'Vukovarsko-Srijemska':   return {color:
colorStation1,opacity: 1, weight: 5, dashArray: '5',fillOpacity: 1};

                            case 'Brodsko-Posavska':   return {color: colorStation1, opacity:
1, weight: 5, dashArray: '5',fillOpacity: 1};

                            case 'Viroviticko-Podravska':   return {color: colorStation1,
opacity: 1, weight: 5, dashArray: '5',fillOpacity: 1};


                            //Stanica2

                            case 'Sisacko-Moslavacka':   return {color: colorStation2, opacity:
1, weight: 5, dashArray: '5',fillOpacity: 1};

                            case 'Bjelovarska-Bilogorska':   return {color: colorStation2,
opacity: 1, weight: 5, dashArray: '5',fillOpacity: 1};

                            case 'Zagrebacka':   return {color: colorStation2, opacity: 1,
weight: 5, dashArray: '5',fillOpacity: 1};

                            case 'Grad Zagreb':   return {color: colorStation2, opacity: 1,
weight: 5, dashArray: '5',fillOpacity: 1};


                            //Stanica3

                            case 'Krapinsko-Zagorska':   return {color: colorStation3, opacity:
1, weight: 5, dashArray: '5',fillOpacity: 1};

                            case 'VaraBћdinska':   return {color: colorStation3, opacity: 1,
weight: 5, dashArray: '5',fillOpacity: 1};

                            case 'Medimurska':   return {color: colorStation3, opacity: 1,
weight: 5, dashArray: '5',fillOpacity: 1};

                            case 'Koprivnicko-KriBћevacka':   return {color: colorStation3,
opacity: 1, weight: 5, dashArray: '5',fillOpacity: 1};


                            //Stanica4

                            case 'Karlovacka':   return {color: colorStation4, opacity: 1,
weight: 5, dashArray: '5',fillOpacity: 1};

                            case 'Licko-Senjska':   return {color: colorStation4, opacity: 1,
weight: 5, dashArray: '5',fillOpacity: 1};

                        case 'Primorsko-Goranska':   return {color: colorStation4, opacity: 1,
weight: 5, dashArray: '5',fillOpacity: 1};

                        case 'Istarska':   return {color: colorStation4, opacity: 1, weight: 5,
dashArray: '5',fillOpacity: 1};


                        //Ostalo

                        case 'Dubrovacko-Neretvanska': return { color: colorRest};

                        case 'Zadarska':   return {color: colorRest};

                        case 'BЈibensko-Kninska':   return {color: colorRest};

                        case 'Splitsko-Dalmatinska':   return {color: colorRest};

                    }

                }


                function createMap(){

                       //zoom disable

                       map.touchZoom.disable();
```

```
        map.doubleClickZoom.disable();
        map.scrollWheelZoom.disable();
        map.boxZoom.disable();
        map.keyboard.disable();
        //map.dragging.disable();
        $(".leaflet-control-zoom").css("visibility", "hidden");
        croatiaLayer = L.geoJson(croatia, {style: style}).addTo(map);

        map.fitBounds(croatiaLayer.getBounds());

        var point1 = [45.45667, 18.01667 ];
        var point2 = [45.36667, 16.51667 ];
        var point3 = [46.19667, 16.51667 ];
        var point4 = [45.16667, 15.11667 ];

        position1 = L.marker(point1).addTo(map).on('click', onClick);
position2 = L.marker(point2).addTo(map).on('click', onClick);
        position3 = L.marker(point3).addTo(map).on('click', onClick);
        position4 = L.marker(point4).addTo(map).on('click', onClick);
}


function dataForPreviousDay()
{
        var station1Avg = 0;
        var station2Avg = 0;
        var station3Avg = 0;
        var station4Avg = 0;
        console.log("test");
        $.ajax({
   type: "POST",
   url: "php/getDataByPreviousDay.php",
   dataType: "json",
   async: false,
   data: {par: par },
   success: function(data)
   {
console.log(data);
                for(i=0; i<data.length; i++)
                {
                        if(i<24)
```

```
                                        {
                                                station1Avg = station1Avg + parseFloat(data[i].par);
                                        }
                                        else if(i>=24 && i<48)
                                        {
                                                station2Avg += parseFloat(data[i].par);
                                        }
                                        else if(i>=48 && i<72)
                                        {
                                                station3Avg += parseFloat(data[i].par);
                                        }
                                        else
                                        {
                                                station4Avg += parseFloat(data[i].par);
                                        }
                                }

                                    averages.push(station1Avg/24);
                                    averages.push(station2Avg/24);
                                    averages.push(station3Avg/24);
                                    averages.push(station4Avg/24);

                           }//end of success
                        });//end of ajax
                       }

                       function setLegend(){
                               var legend = L.control({position: 'bottomright'});
                               legend.onAdd = function (map)
                               {
                                   var div = L.DomUtil.create('div', 'info legend');
                                   labels = [];
                                   if(par == "sm1" || par == "sm2" || par == "sm3" || par == "sm4" || par
== "sm5"
                                       || par == "sm6" || par == "ah1")
                                   {
                                        div.innerHTML += "<p>Vrijednosti u %</p>";
                                           for (var i = 0; i < colorBuffer.length; i++) {
                                               div.innerHTML +=
                                                   '<i style="background-color:' + colorBuffer[i]
+'"></i>' + i*10 + ' &ndash; ' + (i+1)*10 + '<br>';
                                           }
```

```
                        }
                        else if(par == "at1")
                        {
                            div.innerHTML += "<p>Vrijednosti u °C</p>";
                            for (var i = 0; i < colorBuffer.length; i++) {
                                    div.innerHTML +=
                                        '<i style="background-color:' + colorBuffer[i]
+'"></i>' + (-20 +(i*10)) + ' &ndash; ' + ((-20)+(i+1)*10) + '<br>';
                                }
                        }
                        else if(par =="ws1")
                        {
                            div.innerHTML += "<p>Vrijednosti u km/h </p>";
                                for (var i = 0; i < colorBuffer.length; i++) {
                                    div.innerHTML +=
                                        '<i style="background-color:' + colorBuffer[i]
+'"></i>' + i*10 + ' &ndash; ' + (i+1)*10 + '<br>';
                                }
                        }
                        else if(par = "pp1")
                        {
                            div.innerHTML += "<p>Vrijednosti u mm </p>";
                                for (var i = 0; i < colorBuffer.length; i++) {
                                    if(i == 0)
                                    {
                                        div.innerHTML +=
                                        '<i style="background-color:' + colorBuffer[i]
+'"></i>' + i + '<br>';
                                    }
                                    else if(i>0 && i<11)
                                    {
                                        div.innerHTML +=
                                        '<i style="background-color:' + colorBuffer[i]
+'"></i>' + i*5 + ' &ndash; ' + (i+1)*5 + '<br>';
                                    }
                                    else
                                    {
                                        div.innerHTML +=
                                        '<i style="background-color:' + colorBuffer[i]
+'"></i>' + 55 + ' &ndash; ' + 100 + '<br>';
                                    }

                                }
```

```
                }



                return div;

        }


        legend.addTo(map);

        }


        function onClick(e) {


                var positionNumber;

                var par1;


                par1 = $("#map-par").val();

                console.log("Ovo je parametar " + par1);

                var parText = $("#map-par option:selected").text();


                console.log("partext je" + parText);

                if(45.45667 == e.latlng.lat){ positionNumber = 1; tableString ="<h3>Stanica
1</h3><hr>"}

                else if(45.36667 == e.latlng.lat){ positionNumber = 2;  tableString
="<h3>Stanica 2</h3><hr>"}

                else if(46.19667 == e.latlng.lat) {positionNumber = 3;  tableString
="<h3>Stanica 3</h3><hr>"}

                else{ positionNumber = 4; tableString ="<h3>Stanica 4</h3><hr>"}



                $.ajax({

           type: "POST",

           url: "php/getDataByStation.php",

           dataType: "json",

           data: {stationID: positionNumber, par: par1 },

           success: function(data)

           {

        console.log("uspjeh");

        console.log(data);

        tableString = tableString + "<h4>Prikaz mjerenja za prethodnih 7 dana za par " +
parText +"</h4><hr><div><table class='table table-bordered map-
table'><tr><td></td><th>00:00</th><th>01:00</th><th>02:00</th><th>03:00</th><th>04:00</th><th>05:
00</th><th>06:00</th><th>07:00</th><th>08:00</th><th>09:00</th><th>10:00</th><th>11:00</th><th>12
:00</th><th>13:00</th><th>14:00</th><th>15:00</th><th>16:00</th><th>17:00</th><th>18:00</th><th>1
9:00</th><th>20:00</th><th>21:00</th><th>22:00</th><th>23:00</th></tr>";

        var tableData=" ";

                        var buffer= [];
```

```
                var counter = 0;


//SM1 ROW
for(i=0; i<data.length-1;i++)
{
        console.log(i);
        if(i%24 == 0)
        {
                        buffer.push({
                        datum: data[i].datum,
                        v00: data[i].par,
                        v01: data[i+1].par,
                        v02: data[i+2].par,
                        v03: data[i+3].par,
                        v04: data[i+4].par,
                        v05: data[i+5].par,
                        v06: data[i+6].par,
                        v07: data[i+7].par,
                        v08: data[i+8].par,
                        v09: data[i+9].par,
                        v10: data[i+10].par,
                        v11: data[i+11].par,
                        v12: data[i+12].par,
                        v13: data[i+13].par,
                        v14: data[i+14].par,
                        v15: data[i+15].par,
                        v16: data[i+16].par,
                        v17: data[i+17].par,
                        v18: data[i+18].par,
                        v19: data[i+19].par,
                        v20: data[i+20].par,
                        v21: data[i+21].par,
                        v22: data[i+22].par,
                        v23: data[i+23].par
                })
        }
}


for(i=0; i<buffer.length;i++)
{
        console.log(buffer[i].datum.slice(0,10));
```

```
                        console.log(buffer[i].v00);

                        tableData +=

                        "<tr><th class='left-th'style='width:300px'>" + buffer[i].datum.slice(0,10)
+ "</th><td>" + buffer[i].v00 +

                        "</td><td>"  + buffer[i].v01 + "</td><td>" + buffer[i].v02 + "</td><td>"+
buffer[i].v03 + "</td><td>" + buffer[i].v04 + "</td><td>"+ buffer[i].v05 + "</td><td>" +
buffer[i].v06 + "</td><td>"+ buffer[i].v07 + "</td><td>"+ buffer[i].v08 + "</td><td>"+
buffer[i].v09 + "</td><td>"+ buffer[i].v10 + "</td><td>"+ buffer[i].v11 + "</td><td>"+
buffer[i].v12 + "</td><td>"+ buffer[i].v13 + "</td><td>"+ buffer[i].v14 + "</td><td>"+
buffer[i].v15 + "</td><td>"+ buffer[i].v16 + "</td><td>"+ buffer[i].v17 + "</td><td>"+
buffer[i].v18 + "</td><td>"+ buffer[i].v19 + "</td><td>"+ buffer[i].v20 + "</td><td>"+
buffer[i].v21 + "</td><td>"+ buffer[i].v22 + "</td><td>" + buffer[i].v23 + "</td><tr>"



                }



                tableData += "</table>";

        tableString = tableString + tableData;



                if(positionNumber == 1){ position1.bindPopup(tableString,{maxWidth:
"auto"}).openPopup(); }

        else if(positionNumber == 2){ position2.bindPopup(tableString,{maxWidth:
"auto"}).openPopup(); }

        else if(positionNumber == 3){ position3.bindPopup(tableString,{maxWidth:
"auto"}).openPopup(); }

        else if(positionNumber == 4){ position4.bindPopup(tableString,{maxWidth:
"auto"}).openPopup(); }



                   },//end of success

                    error: function (request, status, error) {

                                console.log(request.responseText);

                                console.log("errr");

                          }



                });//end of ajax

                }



                function detectParametarChange(){

                        $('#map-par').on('change', function() {

                                par = $(this).val();

                                console.log("parametar change " + par);

                                averages = [];

                                dataForPreviousDay();

                        croatiaLayer.setStyle(style);

                        $( ".info" ).remove();

                        setLegend();

                        setInfo();

                        });}
```

secondGraphicDisplay.js

```
            var secondCounter = 0;

        var startDate;

        var endDate;

        var par;


        $(document).ready(function() {


            $('#reportrange').daterangepicker(

            {

              autoUpdateInput: false // da mi automatski ne stavi vrijednost

            },

            function(start, end) {

              $('#reportrange').val(start.format('YYYY/MM/DD') + " - " +
end.format('YYYY/MM/DD')); // set value to input element

            }

            );



            $('#save').click(function(){

              par = $("#parametar").val();

              var enteredDate= $("#reportrange").val(); // uzimanje vrijednosti

              //provjera dali postoji vrijednost,ako ne izbaci poruku

              if(!enteredDate) {

               $('#validationAlert').modal('show');

              }

              else

              {


                //split entered  date to startDate and endDate

                var splitedDateContainer = enteredDate.split('-');

                //get splited and trimed date

                var startDate = $.trim(splitedDateContainer[0]);

                var endDate = $.trim(splitedDateContainer[1]);


                $.ajax({

                  type: "POST",

                  url: "php/getDataByPar.php",

                  dataType: "json",

                  data: {startDate: startDate, endDate: endDate, parametar: par},

                  success: function(data) {
```

```
                console.log(data);
                     //provjerava data ima za property numRowsError
                     if(data.hasOwnProperty('numRowsError')){
                        console.log("num rows error");
                       $('#noDataAlert').modal('show');
                     }else{
                       secondCounter++;
                       waitingDialog.show('Graf se učitava...', {dialogSize: 'sm', progressType:
'warning'});
                       setTimeout(function () {
                         waitingDialog.hide();
                         showGraph(data, secondCounter);
                       }, 1500);
                     }

                   },//end of success
                   error: function (request, status, error) {
                console.log(request.responseText);

                    }
                 });//end of ajax

               }//end of if(!enteredDate)

            });//end of $('#save').click(function(){

          });//end of $(document).ready(function()


        function showGraph(data, secondCounter) {

          var buffer = [];
          var current;
          var date;
          var counter = 0;

                for(var i = 0; i < data.length; i++)
                {
                        //console.log(i);

                        date = new Date(data[i][0]);
```

```
                //prvu iteraciju postavi prvi date na current
                if(i < 1)
                {
                    current = date;
                    buffer[counter] = [];
                }
                //console.log("current" + current);
                //console.log("date" + date);
                //provjerava dali je date isti

                if(current.toString() != date.toString())
                {
                    //console.log("razliciti");
                    //console.log("counter" + counter);
                    current = date;
                    counter++;
                    buffer[counter] = [];
                }

                buffer[counter][0]=current; // spremanje datea na prvo mjesto
                buffer[counter][data[i][1]] = data[i][2]; // spremanje vrijednosti na mjesto
1, 2, 3, ili 4
            }

        console.log(buffer);
        //var startDate = arr[1][0];
        //var endDate = arr[test.length][0];

        var startDate = dateFormat(buffer[0][0], "yyyy/dd/mm");
        var endDate = dateFormat(buffer[buffer.length-1][0], "yyyy/dd/mm");

        var choosenPar;
        switch (par) {
            case "sm1":
                choosenPar = "Vlažnost tla 1";
                break;
            case "sm2":
                choosenPar = "Vlažnost tla 2";
                break;
            case "sm3":
                choosenPar = "Vlažnost tla 3";
```

```
                break;
            case "sm4":
                choosenPar = "Vlažnost tla 4";
                break;
            case "sm5":
                choosenPar = "Vlažnost tla 5";
                break;
            case "sm6":
                choosenPar = "Vlažnost tla 6";
                break;
            case "ah1":
                choosenPar = "Vlažnost zraka";
                break;
         case "at1":
        choosenPar = "Temperatura zraka";
        break;
            case "ws1":
                choosenPar = "Brzina vjetra";
                break;
            case "pp1":
                choosenPar = "Padaline";
                break;
        }


        // provjerava dali se objekt kreira prvi put, ako se ne kreira onda samo update napravi sa
datama
        if(secondCounter == 1)
        {
          g2 = new Dygraph(
            document.getElementById("graphdiv2"),
            buffer,
             {
                labels: [ "date", "Stanica1", "Stanica2", "Stanica3", "Stanica4"],
                legend: 'always',
                title: 'Prikaz mjerenja za parametar ' + choosenPar + ' u periodu od ' + startDate
+ ' do ' + endDate,
                titleHeight: 32,
                ylabel: 'Vrijednosti',
                xlabel: 'Vremenski period',
             }
          );
        }
```

```
        else
        {
          g2.updateOptions(
              {
                title: 'Prikaz mjerenja za parametar ' + choosenPar + ' u periodu od ' + startDate
+ ' do ' + endDate,
                'file': buffer
              }
          );
        }
        }

        function sleep(miliseconds) {
           var currentTime = new Date().getTime();
           while (currentTime + miliseconds >= new Date().getTime()) {
           }
        }
```

Connect.php

```
<?php
  // Connect to MySQL
  $mysqli = new mysqli( "localhost", "root", "", "dipl");

  // Check our connection
  if ( $mysqli->connect_error ) {
    die( 'Connect Error: ' . $mysqli->connect_errno . ': ' . $mysqli->connect_error );
  }
```

getDataByPar.php

```
<?php

require 'connect.php';

$startDate = $_POST['startDate'];

$eD = $_POST['endDate'];

$par = $_POST['parametar'];

$endDate=strftime("%Y/%m/%d", strtotime("$eD +1 day"));



if ($result = $mysqli->query("SELECT datum, stanica_id, {$par} FROM podaci WHERE datum between
'$startDate'and '$endDate' ORDER BY datum ASC"))

{

        $numRows = $result->num_rows;


        if($numRows>0){


                //loop through selected data

                while($row = $result->fetch_row()) {

                        $respond[]=$row;

                }

        }else{

                $respond['numRowsError'] = 0;

        }



}else{

        die("QUERY ERROR: " . $result->error);

}


echo json_encode($respond);
```

getDataByPreviousDay.php

```
<?php

require 'connect.php';

$par = $_POST['par'];


if ($result =  $mysqli->query("SELECT stanica_id, datum, {$par} FROM podaci WHERE datum between
subdate(CURDATE(), 1) and CURDATE() ORDER BY stanica_id"))

{

        $numRows = $result->num_rows;

        if($numRows>0)

        {

                $counter = 0;

                //loop through selected data

                while($row = $result->fetch_row())

                {

                        $respond[$counter]["stanica"] = $row[0];

                        $respond[$counter]["datum"] = $row[1];

                        $respond[$counter]["par"] = $row[2];

                        $counter++;

                }

        }

        else

        {

                $respond['numRowsError'] = 0;

        }

}

else

{

        die("QUERY ERROR: " . $result->error);

}


echo json_encode($respond);
```

getDataByStation.php

```
<?php

require 'connect.php';

$stationID = $_POST['stationID'];

$par = $_POST['par'];


if ($result =  $mysqli->query("SELECT datum, {$par} FROM podaci WHERE datum between
subdate(CURDATE(), 7) and CURDATE() AND stanica_id = '$stationID' ORDER BY datum ASC"))

{

        $numRows = $result->num_rows;


        if($numRows>0)

        {

                $counter = 0;

                //loop through selected data

                while($row = $result->fetch_row())

                {

                        $respond[$counter]["datum"] = $row[0];

                        $respond[$counter]["par"] = $row[1];

                        $counter++;

                }

        }

        else

        {

                $respond['numRowsError'] = 0;

        }




}

else

{

        die("QUERY ERROR: " . $result->error);

}


echo json_encode($respond);
```

style.css

```
body
{
	margin:0; /* This is used to reset any browser-default margins */
}

#map
{
	height:100vh;
	margin-bottom: 120px;
}

.app-wrapper{
	padding: 120px 15px;
    text-align: center;
}

#graphdiv2{
	margin-bottom: 200px;
}



.dygraph-legend{
	 width: auto !important;
	 top: 550px !important;
	 left: 50px !important;
}

.infotext { }

.dygraph-label { font-family: Georgia, Verdana, serif; }

.dygraph-title { font-size: 20px; text-shadow: gray 1px 1px 1px; }

.dygraph-ylabel { font-size: 18px; text-shadow: gray -2px 2px 2px; }
.dygraph-xlabel { font-size: 18px; text-shadow: gray -2px 2px 2px; }
.chart { border: 1px dashed black; margin: 5px 5px 5px 50px; padding: 2px; }

.map-table{
	 white-space: nowrap;
}
```

```
.map-table th
{
        font-size: 10px;
        padding: 5px !important;
}

.map-table td
{
    font-size: 10px;
    text-align: center;
    padding: 5px !important;
}
.leaflet-popup-content
{
    margin: 45px !important;
}

.legend{

    color: #555;
    background-color: blue;
}

.info{
        padding: 6px 8px;
    font: 14px/16px Arial, Helvetica, sans-serif;
    background: white;
    background: rgba(255,255,255,0.8);
    box-shadow: 0 0 15px rgba(0,0,0,0.2);
    border-radius: 5px;
    line-height: 20px;
}


.legend i {
    width: 18px;
    height: 18px;
    float: left;
    margin-right: 8px;
    opacity: 0.7;
    clear: both;
```

```
}

.station-desc{
    text-align: left;
}
```

Index.php

```
<?php

require('connect.php');

require('insertData.php');



$mjerenje1 = '2016-01-01 00:00:00';

echo '<b>' . "Datum************" . '**** ' . "SM1" . ' **** ' . "SM2" . ' **** ' . "SM3". ' ****
' . "SM4" . ' **** ' . "SM5" . ' **** ' . "SM6" . ' **** ' . "AH1" . ' **** ' . "AT1" . ' **** '
. "PP1" . ' **** ' . "WS1"  . ' **** ' . "WD1" .'</b><br>';


//petlja

for ($i = 0; $i < 365; $i++)

{

        $sm1base = rand(0, 100);

        $sm2base = rand(0, 100);

        $sm3base = rand(0, 100);

        $sm4base = rand(0, 100);

        $sm5base = rand(0, 100);

        $sm6base = rand(0, 100);

        $ah1base = rand(0, 100);

        $at1base = rand(-20, 40);

        $pp1base = rand(0, 100);

        $ws1base = rand(0, 40);

        $wd1base = rand(0, 360);


    $mjerenje1  = date('Y-m-d H:i:s', strtotime($mjerenje1. ' +1 day'));

    $mjerenje2 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+1 hours'));

    $mjerenje3 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+2 hours'));

    $mjerenje4 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+3 hours'));

    $mjerenje5 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+4 hours'));

    $mjerenje6 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+5 hours'));

    $mjerenje7 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+6 hours'));

    $mjerenje8 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+7 hours'));

    $mjerenje9 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+8 hours'));

    $mjerenje10 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+9 hours'));

    $mjerenje11 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+10 hours'));

    $mjerenje12 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+11 hours'));

    $mjerenje13 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+12 hours'));

    $mjerenje14 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+13 hours'));

    $mjerenje15 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+14 hours'));

    $mjerenje16 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+15 hours'));
```

```
$mjerenje17 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+16 hours'));
$mjerenje18 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+17 hours'));
$mjerenje19 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+18 hours'));
$mjerenje20 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+19 hours'));
$mjerenje21 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+20 hours'));
$mjerenje22 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+21 hours'));
$mjerenje23 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+22 hours'));
$mjerenje24 = date('Y-m-d H:i:s', strtotime($mjerenje1. '+23 hours'));

$sm1_1 = number_format((float)$sm1base, 2, '.', '') ;
    $sm1_2 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_3 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_4 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_5 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_6 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_7 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_8 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_9 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_10 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_11 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_12 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_13 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_14 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_15 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_16 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_17 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_18 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_19 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_20 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_21 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_22 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_23 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm1_24 = number_format((float)$sm1base +  (rand(0, 10) / 10), 2, '.', '') ;

$sm2_1 = number_format((float)$sm2base, 2, '.', '') ;
$sm2_2 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm2_3 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm2_4 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm2_5 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm2_6 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm2_7 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
```

```
$sm2_8 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm2_9 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm2_10 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm2_11 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm2_12 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm2_13 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm2_14 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm2_15 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm2_16 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm2_17 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm2_18 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm2_19 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm2_20 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm2_21 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm2_22 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm2_23 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm2_24 = number_format((float)$sm2base +  (rand(0, 10) / 10), 2, '.', '') ;

$sm3_1 = number_format((float)$sm3base, 2, '.', '');
    $sm3_2 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm3_3 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm3_4 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm3_5 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm3_6 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm3_7 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm3_8 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm3_9 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm3_10 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm3_11 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm3_12 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm3_13 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm3_14 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm3_15 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm3_16 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm3_17 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm3_18 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm3_19 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm3_20 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm3_21 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm3_22 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
    $sm3_23 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;
```

```
$sm3_24 = number_format((float)$sm3base +  (rand(0, 10) / 10), 2, '.', '') ;


$sm4_1 = number_format((float)$sm4base, 2, '.', '');
$sm4_2 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_3 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_4 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_5 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_6 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_7 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_8 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_9 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_10 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_11 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_12 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_13 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_14 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_15 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_16 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_17 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_18 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_19 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_20 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_21 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_22 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_23 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm4_24 = number_format((float)$sm4base +  (rand(0, 10) / 10), 2, '.', '') ;


$sm5_1 = number_format((float)$sm5base, 2, '.', '');
$sm5_2 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm5_3 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm5_4 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm5_5 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm5_6 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm5_7 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm5_8 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm5_9 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm5_10 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm5_11 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm5_12 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm5_13 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm5_14 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
```

```
$sm5_15 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm5_16 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm5_17 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm5_18 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm5_19 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm5_20 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm5_21 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm5_22 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm5_23 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm5_24 = number_format((float)$sm5base +  (rand(0, 10) / 10), 2, '.', '') ;

$sm6_1 = number_format((float)$sm6base, 2, '.', '');
$sm6_2 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_3 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_4 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_5 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_6 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_7 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_8 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_9 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_10 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_11 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_12 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_13 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_14 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_15 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_16 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_17 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_18 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_19 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_20 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_21 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_22 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_23 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;
$sm6_24 = number_format((float)$sm6base +  (rand(0, 10) / 10), 2, '.', '') ;

$ah1_1 = number_format((float)$ah1base, 2, '.', '');
$ah1_2 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ah1_3 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ah1_4 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ah1_5 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
```

```
$ah1_6 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ah1_7 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ah1_8 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ah1_9 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ah1_10 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ah1_11 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ah1_12 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ah1_13 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ah1_14 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ah1_15 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ah1_16 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ah1_17 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ah1_18 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ah1_19 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ah1_20 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ah1_21 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ah1_22 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ah1_23 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ah1_24 = number_format((float)$ah1base +  (rand(0, 10) / 10), 2, '.', '') ;

$at1_1 = number_format((float)$at1base, 2, '.', '');
$at1_2 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
$at1_3 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
$at1_4 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
$at1_5 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
$at1_6 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
$at1_7 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
$at1_8 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
$at1_9 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
$at1_10 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
$at1_11 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
$at1_12 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
$at1_13 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
$at1_14 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
$at1_15 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
$at1_16 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
$at1_17 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
$at1_18 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
$at1_19 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
$at1_20 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
$at1_21 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
```

```
$at1_22 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
$at1_23 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;
$at1_24 = number_format((float)$at1base +  (rand(0, 10) / 10), 2, '.', '') ;

//svaku petu ce staviti padalinu
if($i%5 == 0)
{
        $pp1_1 = number_format((float)$pp1base, 2, '.', '');
        $pp1_2 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_3 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_4 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_5 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_6 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_7 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_8 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_9 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_10 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_11 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_12 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_13 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_14 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_15 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_16 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_17 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_18 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_19 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_20 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_21 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_22 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_23 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
        $pp1_24 = number_format((float)$pp1base +  (rand(0, 10) / 10), 2, '.', '') ;
}
else
{
 $pp1_1 = 0;
        $pp1_2 = 0;
        $pp1_3 = 0;
        $pp1_4 = 0;
        $pp1_5 = 0;
        $pp1_6 = 0;
        $pp1_7 = 0;
```

```
        $pp1_8 = 0;
        $pp1_9 = 0;
        $pp1_10 = 0;
        $pp1_11 = 0;
        $pp1_12 = 0;
        $pp1_13 = 0;
        $pp1_14 = 0;
        $pp1_15 = 0;
        $pp1_16 = 0;
        $pp1_17 = 0;
        $pp1_18 = 0;
        $pp1_19 = 0;
        $pp1_20 = 0;
        $pp1_21 = 0;
        $pp1_22 = 0;
        $pp1_23 = 0;
        $pp1_24 = 0;

}


$ws1_1 = number_format((float)$ws1base, 2, '.', '');
$ws1_2 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ws1_3 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ws1_4 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ws1_5 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ws1_6 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ws1_7 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ws1_8 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ws1_9 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ws1_10 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ws1_11 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ws1_12 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ws1_13 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ws1_14 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ws1_15 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ws1_16 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ws1_17 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ws1_18 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ws1_19 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ws1_20 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
```

```
$ws1_21 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ws1_22 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ws1_23 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;
$ws1_24 = number_format((float)$ws1base +  (rand(0, 10) / 10), 2, '.', '') ;



$wd1_1 = number_format((float)$wd1base, 2, '.', '');
$wd1_2 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_3 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_4 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_5 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_6 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_7 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_8 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_9 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_10 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_11 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_12 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_13 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_14 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_15 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_16 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_17 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_18 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_19 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_20 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_21 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_22 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_23 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
$wd1_24 = number_format((float)$wd1base +  (rand(0, 10) / 10), 2, '.', '') ;
```

```
      echo $mjerenje1 . ' **** ' . $sm1_1 . ' **** ' . $sm2_1 . ' **** ' . $sm3_1 . ' **** ' .
$sm4_1 . ' **** ' . $sm5_1 . ' **** ' . $sm6_1 . ' **** ' . $ah1_1. ' **** ' . $at1_1 . ' **** '
. $pp1_1 . ' **** ' . $ws1_1 . ' **** ' . $wd1_1 .'<br>' .
```

```
              $mjerenje2 . ' **** ' . $sm1_2 . ' **** ' . $sm2_2 . ' **** ' . $sm3_2 . ' **** '
. $sm4_2 . ' **** ' . $sm5_2 . ' **** ' . $sm6_2 . ' **** ' . $ah1_2. ' **** ' . $at1_2 . ' ****
' . $pp1_2 . ' **** ' . $ws1_2 . ' **** ' . $wd1_2 .'<br>' .
```

```
              $mjerenje3 . ' **** ' . $sm1_3 . ' **** ' . $sm2_3 . ' **** ' . $sm3_3 . ' **** '
. $sm4_3 . ' **** ' . $sm5_3 . ' **** ' . $sm6_3 . ' **** ' . $ah1_3. ' **** ' . $at1_3. ' **** '
. $pp1_3 . ' **** ' . $ws1_3 . ' **** ' . $wd1_3 .'<br>' .
```

```
                $mjerenje4 . ' **** ' . $sm1_4 . ' **** ' . $sm2_4 . ' **** ' . $sm3_4 . ' **** '
. $sm4_4 . ' **** ' . $sm5_4 . ' **** ' . $sm6_4 . ' **** ' . $ah1_4. ' **** ' . $at1_4 . ' ****
' . $pp1_4 . ' **** ' . $ws1_4 . ' **** ' . $wd1_4 .'<br>' .


                $mjerenje5 . ' **** ' . $sm1_5 . ' **** ' . $sm2_5 . ' **** ' . $sm3_5 . ' **** '
. $sm4_5 . ' **** ' . $sm5_5 . ' **** ' . $sm6_5 . ' **** ' . $ah1_5. ' **** ' . $at1_5 . ' ****
' . $pp1_5 . ' **** ' . $ws1_5 . ' **** ' . $wd1_5 .'<br>' .




                $mjerenje6 . ' **** ' . $sm1_6 . ' **** ' . $sm2_6 . ' **** ' . $sm3_6 . ' **** '
. $sm4_6 . ' **** ' . $sm5_6 . ' **** ' . $sm6_6 . ' **** ' . $ah1_6. ' **** ' . $at1_6 . ' ****
' . $pp1_6 . ' **** ' . $ws1_6 . ' **** ' . $wd1_6 .'<br>' .


                $mjerenje7 . ' **** ' . $sm1_7 . ' **** ' . $sm2_7 . ' **** ' . $sm3_7 . ' **** '
. $sm4_7 . ' **** ' . $sm5_7 . ' **** ' . $sm6_7 . ' **** ' . $ah1_7. ' **** ' . $at1_7 . ' ****
' . $pp1_7 . ' **** ' . $ws1_7 . ' **** ' . $wd1_7 .'<br>' .




                $mjerenje8 . ' **** ' . $sm1_8 . ' **** ' . $sm2_8 . ' **** ' . $sm3_8 . ' **** '
. $sm4_8 . ' **** ' . $sm5_8 . ' **** ' . $sm6_8 . ' **** ' . $ah1_8. ' **** ' . $at1_8 . ' ****
' . $pp1_8 . ' **** ' . $ws1_8 . ' **** ' . $wd1_8 .'<br>' .


                $mjerenje9 . ' **** ' . $sm1_9 . ' **** ' . $sm2_9 . ' **** ' . $sm3_9 . ' **** '
. $sm4_9 . ' **** ' . $sm5_9 . ' **** ' . $sm6_9 . ' **** ' . $ah1_9. ' **** ' . $at1_9 . ' ****
' . $pp1_9 . ' **** ' . $ws1_9 . ' **** ' . $wd1_9 .'<br>' .


                $mjerenje10 . ' **** ' . $sm1_10 . ' **** ' . $sm2_10 . ' **** ' . $sm3_10 . '
**** ' . $sm4_10 . ' **** ' . $sm5_10 . ' **** ' . $sm6_10 . ' **** ' . $ah1_10. ' **** ' .
$at1_10 . ' **** ' . $pp1_10 . ' **** ' . $ws1_10 . ' **** ' . $wd1_10 .'<br>' .


                $mjerenje11 . ' **** ' . $sm1_11 . ' **** ' . $sm2_11 . ' **** ' . $sm3_11 . '
**** ' . $sm4_11 . ' **** ' . $sm5_11 . ' **** ' . $sm6_11 . ' **** '  . $ah1_11. ' **** ' .
$at1_11 . ' **** ' . $pp1_11 . ' **** ' . $ws1_11 . ' **** ' . $wd1_11 .'<br>' .


                $mjerenje12 . ' **** ' . $sm1_12 . ' **** ' . $sm2_12 . ' **** ' . $sm3_12. '
**** ' . $sm4_12 . ' **** ' . $sm5_12 . ' **** ' . $sm6_12 . ' **** ' . $ah1_12. ' **** ' .
$at1_12 . ' **** ' . $pp1_12 . ' **** ' . $ws1_12 . ' **** ' . $wd1_12 .'<br>' .


                $mjerenje13 . ' **** ' . $sm1_13 . ' **** ' . $sm2_13 . ' **** ' . $sm3_13 . '
**** ' . $sm4_13 . ' **** ' . $sm5_13 . ' **** ' . $sm6_13 . ' **** ' . $ah1_13. ' **** ' .
$at1_13 . ' **** ' . $pp1_13 . ' **** ' . $ws1_13 . ' **** ' . $wd1_13 .'<br>' .


                $mjerenje14 . ' **** ' . $sm1_14 . ' **** ' . $sm2_14 . ' **** ' . $sm3_14 . '
**** ' . $sm4_14 . ' **** ' . $sm5_14 . ' **** ' . $sm6_14 . ' **** ' . $ah1_14. ' **** ' .
$at1_14 . ' **** ' . $pp1_14 . ' **** ' . $ws1_14 . ' **** ' . $wd1_14 .'<br>' .


                $mjerenje15 . ' **** ' . $sm1_15 . ' **** ' . $sm2_15 . ' **** ' . $sm3_15 . '
**** ' . $sm4_15 . ' **** ' . $sm5_15 . ' **** ' . $sm6_15 . ' **** ' . $ah1_15. ' **** ' .
$at1_15 . ' **** ' . $pp1_15 . ' **** ' . $ws1_15 . ' **** ' . $wd1_15 .'<br>' .
```

```
                $mjerenje16 . ' **** ' . $sm1_16 . ' **** ' . $sm2_16 . ' **** ' . $sm3_16 . '
**** ' . $sm4_16 . ' **** ' . $sm5_16 . ' **** ' . $sm6_16 . ' **** '. $ah1_16. ' **** ' .
$at1_16 . ' **** ' . $pp1_16 . ' **** ' . $ws1_16 . ' **** ' . $wd1_16 .'<br>' .


                $mjerenje17 . ' **** ' . $sm1_17 . ' **** ' . $sm2_17 . ' **** ' . $sm3_17 . '
**** ' . $sm4_17 . ' **** ' . $sm5_17 . ' **** ' . $sm6_17 . ' **** ' . $ah1_17. ' **** ' .
$at1_17 . ' **** ' . $pp1_17 . ' **** ' . $ws1_17 . ' **** ' . $wd1_17 .'<br>' .


                $mjerenje18 . ' **** ' . $sm1_18 . ' **** ' . $sm2_18 . ' **** ' . $sm3_18 . '
**** ' . $sm4_18 . ' **** ' . $sm5_18 . ' **** ' . $sm6_18 . ' **** ' . $ah1_18. ' **** ' .
$at1_18 . ' **** ' . $pp1_18 . ' **** ' . $ws1_18 . ' **** ' . $wd1_18 .'<br>' .


                $mjerenje19 . ' **** ' . $sm1_19 . ' **** ' . $sm2_19 . ' **** ' . $sm3_19 . '
**** ' . $sm4_19 . ' **** ' . $sm5_19 . ' **** ' . $sm6_19 . ' **** ' . $ah1_19. ' **** ' .
$at1_19 . ' **** ' . $pp1_19 . ' **** ' . $ws1_19 . ' **** ' . $wd1_19 .'<br>' .


                $mjerenje20 . ' **** ' . $sm1_20 . ' **** ' . $sm2_20 . ' **** ' . $sm3_20 . '
**** ' . $sm4_20 . ' **** ' . $sm5_20 . ' **** ' . $sm6_20 . ' **** ' . $ah1_20. ' **** ' .
$at1_20 . ' **** ' . $pp1_20 . ' **** ' . $ws1_20 . ' **** ' . $wd1_20 .'<br>' .


                $mjerenje21 . ' **** ' . $sm1_21 . ' **** ' . $sm2_21 . ' **** ' . $sm3_21 . '
**** ' . $sm4_21 . ' **** ' . $sm5_21 . ' **** ' . $sm6_21 . ' **** '. $ah1_21. ' **** ' .
$at1_21 . ' **** ' . $pp1_21 . ' **** ' . $ws1_21 . ' **** ' . $wd1_21 .'<br>' .


                $mjerenje22 . ' **** ' . $sm1_22 . ' **** ' . $sm2_22 . ' **** ' . $sm3_22 . '
**** ' . $sm4_22 . ' **** ' . $sm5_22 . ' **** ' . $sm6_22 . ' **** ' . $ah1_22. ' **** ' .
$at1_22 . ' **** ' . $pp1_22 . ' **** ' . $ws1_22 . ' **** ' . $wd1_22 .'<br>' .


                $mjerenje23 . ' **** ' . $sm1_23 . ' **** ' . $sm2_23 . ' **** ' . $sm3_23 . '
**** ' . $sm4_23 . ' **** ' . $sm5_23 . ' **** ' . $sm6_23 . ' **** '  . $ah1_23. ' **** ' .
$at1_23 . ' **** ' . $pp1_23 . ' **** ' . $ws1_23 . ' **** ' . $wd1_23;


                $mjerenje24 . ' **** ' . $sm1_24 . ' **** ' . $sm2_24 . ' **** ' . $sm3_24 . '
**** ' . $sm4_24 . ' **** ' . $sm5_24 . ' **** ' . $sm6_24. ' **** '  . $ah1_24. ' **** ' .
$at1_24 . ' **** ' . $pp1_24 . ' **** ' . $ws1_24 . ' **** ' . $wd1_24;




        echo '<hr>';

insertData($mysqli,$mjerenje1,$sm1_1,$sm2_1,$sm3_1,$sm4_1,$sm5_1,$sm6_1,$at1_1,$ah1_1,$ws1_1,$wd1
_1,$pp1_1);

insertData($mysqli,$mjerenje2,$sm1_2,$sm2_2,$sm3_2,$sm4_2,$sm5_2,$sm6_2,$at1_2,$ah1_2,$ws1_2,$wd1
_2,$pp1_2);

insertData($mysqli,$mjerenje3,$sm1_3,$sm2_3,$sm3_3,$sm4_3,$sm5_3,$sm6_3,$at1_3,$ah1_3,$ws1_3,$wd1
_3,$pp1_3);

insertData($mysqli,$mjerenje4,$sm1_4,$sm2_4,$sm3_4,$sm4_4,$sm5_4,$sm6_4,$at1_4,$ah1_4,$ws1_4,$wd1
_4,$pp1_4);

insertData($mysqli,$mjerenje5,$sm1_5,$sm2_5,$sm3_5,$sm4_5,$sm5_5,$sm6_5,$at1_5,$ah1_5,$ws1_5,$wd1
_5,$pp1_5);
```

```
insertData($mysqli,$mjerenje6,$sm1_6,$sm2_6,$sm3_6,$sm4_6,$sm5_6,$sm6_6,$at1_6,$ah1_6,$ws1_6,$wd1
_6,$pp1_6);


insertData($mysqli,$mjerenje7,$sm1_7,$sm2_7,$sm3_7,$sm4_7,$sm5_7,$sm6_7,$at1_7,$ah1_7,$ws1_7,$wd1
_7,$pp1_7);


insertData($mysqli,$mjerenje8,$sm1_8,$sm2_8,$sm3_8,$sm4_8,$sm5_8,$sm6_8,$at1_8,$ah1_8,$ws1_8,$wd1
_8,$pp1_8);


insertData($mysqli,$mjerenje9,$sm1_9,$sm2_9,$sm3_9,$sm4_9,$sm5_9,$sm6_9,$at1_9,$ah1_9,$ws1_9,$wd1
_9,$pp1_9);


insertData($mysqli,$mjerenje10,$sm1_10,$sm2_10,$sm3_10,$sm4_10,$sm5_10,$sm6_10,$at1_10,$ah1_10,$w
s1_10,$wd1_10,$pp1_10);


insertData($mysqli,$mjerenje11,$sm1_11,$sm2_11,$sm3_11,$sm4_11,$sm5_11,$sm6_11,$at1_11,$ah1_11,$w
s1_11,$wd1_11,$pp1_11);


insertData($mysqli,$mjerenje12,$sm1_12,$sm2_12,$sm3_12,$sm4_12,$sm5_12,$sm6_12,$at1_12,$ah1_12,$w
s1_12,$wd1_12,$pp1_12);


insertData($mysqli,$mjerenje13,$sm1_13,$sm2_13,$sm3_13,$sm4_13,$sm5_13,$sm6_13,$at1_13,$ah1_13,$w
s1_13,$wd1_13,$pp1_13);


insertData($mysqli,$mjerenje14,$sm1_14,$sm2_14,$sm3_14,$sm4_14,$sm5_14,$sm6_14,$at1_14,$ah1_14,$w
s1_14,$wd1_14,$pp1_14);


insertData($mysqli,$mjerenje15,$sm1_15,$sm2_15,$sm3_15,$sm4_15,$sm5_15,$sm6_15,$at1_15,$ah1_15,$w
s1_15,$wd1_15,$pp1_15);


insertData($mysqli,$mjerenje16,$sm1_16,$sm2_16,$sm3_16,$sm4_16,$sm5_16,$sm6_16,$at1_16,$ah1_16,$w
s1_16,$wd1_16,$pp1_16);


insertData($mysqli,$mjerenje17,$sm1_17,$sm2_17,$sm3_17,$sm4_17,$sm5_17,$sm6_17,$at1_17,$ah1_17,$w
s1_17,$wd1_17,$pp1_17);


insertData($mysqli,$mjerenje18,$sm1_18,$sm2_18,$sm3_18,$sm4_18,$sm5_18,$sm6_18,$at1_18,$ah1_18,$w
s1_18,$wd1_18,$pp1_18);


insertData($mysqli,$mjerenje19,$sm1_19,$sm2_19,$sm3_19,$sm4_19,$sm5_19,$sm6_19,$at1_19,$ah1_19,$w
s1_19,$wd1_19,$pp1_19);


insertData($mysqli,$mjerenje20,$sm1_20,$sm2_20,$sm3_20,$sm4_20,$sm5_20,$sm6_20,$at1_20,$ah1_20,$w
s1_20,$wd1_20,$pp1_20);


insertData($mysqli,$mjerenje21,$sm1_21,$sm2_21,$sm3_21,$sm4_21,$sm5_21,$sm6_21,$at1_21,$ah1_21,$w
s1_21,$wd1_21,$pp1_21);


insertData($mysqli,$mjerenje22,$sm1_22,$sm2_22,$sm3_22,$sm4_22,$sm5_22,$sm6_22,$at1_22,$ah1_22,$w
s1_22,$wd1_22,$pp1_22);


insertData($mysqli,$mjerenje23,$sm1_23,$sm2_23,$sm3_23,$sm4_23,$sm5_23,$sm6_23,$at1_23,$ah1_23,$w
s1_23,$wd1_23,$pp1_23);


insertData($mysqli,$mjerenje24,$sm1_24,$sm2_24,$sm3_24,$sm4_24,$sm5_24,$sm6_24,$at1_24,$ah1_24,$w
s1_24,$wd1_24,$pp1_24);

}

?>
```

insertData.php

```
<?php

function insertData($mysqli,$datum,$sm1,$sm2,$sm3,$sm4,$sm5,$sm6,$at1,$ah1,$ws1,$wd1,$pp1)
{

        $insertQuiz = "INSERT INTO
      podaci (
          stanica_id,
          datum,
           sm1,
           sm2,
           sm3,
           sm4,
           sm5,
           sm6,
           at1,
           ah1,
           ws1,
           pp1 )
         VALUES (
          4,
         '{$mysqli->real_escape_string($datum) }',
         '{$mysqli->real_escape_string($sm1) }',
         '{$mysqli->real_escape_string($sm2) }',
         '{$mysqli->real_escape_string($sm3) }',
         '{$mysqli->real_escape_string($sm4) }',
         '{$mysqli->real_escape_string($sm5) }',
         '{$mysqli->real_escape_string($sm6) }',
         '{$mysqli->real_escape_string($at1) }',
         '{$mysqli->real_escape_string($ah1) }',
         '{$mysqli->real_escape_string($ws1) }',
         '{$mysqli->real_escape_string($pp1) }'
         )";

         if(!$mysqli->query($insertQuiz))
         {
             echo $mysqli->error;
         }
         else
            { echo 'uspjeh';}}?>
```