**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni diplomski studij računarstva**

**Informacijske i podatkovne znanosti**

**SVJETSKA I DRŽAVNA STATISTIKA COVID-19**

**Vizualizacija podataka**

**Matej Dmitrović**

**Osijek, 2020.**

Contents

[1. OPIS PROJEKTA 3](#_Toc49106855)

[1.1 Korištene tehnologije 3](#_Toc49106856)

[1.2 Programsko rješenje 4](#_Toc49106857)

[1.2.1 HTML 4](#_Toc49106858)

[1.2.2 JavaScript/D3.js 5](#_Toc49106859)

[1.2.3 CSS 9](#_Toc49106860)

[1.3 Vizualizacija podataka 9](#_Toc49106861)

[1.3.1 Primjeri vizualizacije 9](#_Toc49106862)

[2. KORIŠTENE IZVORI 10](#_Toc49106863)

# 1. OPIS PROJEKTA

COVID-19 iz porodice koronavirusa je već znatno vrijeme aktualna tema cijeloga svijeta. Ovaj virus je utjecao na sve države i njihove stanovnike. Neke države su brzo i ošrto reagirale na širenje virusa, dok neke gotovo ignoriraju ovaj novonastali problem. Zbog značaja ovoga virusa, napravljena je vizualizacija njegovog utjecaja – od toga koliko je ljudi ozdravljeno, do toga koliko je ljudi umrlo. Ova vizualizacija je opisana programskim rješenjem i dizajnerskim rješenjem. Bit će moguće vidjeti kako je jedno rješenje utjecalo na drugo, te kako su ta dva rješenja uopće bila moguća.

## 1.1 Korištene tehnologije

Za vizualizaciju projekta korištene su razne tehnologije u jedinstvu. Prva tehnologija od značaja jest HTML opisni jezik. Pomoću HTML-a definirana je opća struktura stranice, a tako i raposred vizualizacije. HTML također donosi mogućnost korištenje *<svg>* elementa koji je zaslužan za oslikavanje grafova kako bi se informacije mogle u što boljem obliku predoćiti korisniku/klijentu.

Dok HTML definira podjelu stranice, skirptni jezik JavaScript služi za dohvaćanje podataka i izgradnju grafova nad zim podacima. Dok sam JavaScript može dinamično manipulirati elementima stranice, puno je teže raditi kompliciranje zadatake poput iscrtavanja samih grafova. U tu svrhu korištena je biblioteka D3.js koja je specijalizirana za prezentiranje podataka na raznim web-stranicama.

Pomoću JavaScript-a i njene biblioteke D3.js mogu se prikazati razni dijagrami, no za izgled same web-stranice se koristi CSS opisni jezik. CSS je također korišten za definiranje stilova pojedinih elemenata koji su zajednički svim elementima, tako da se dio posla D3.js biblioteke prebacuje (i bolje organizira) u CSS-u.

Zbog potrebe za povezivanjem sa vanjskim resursima, korišten je program Laragon za stvaranje virtualnih servera. Laragon sa sobom donosi PHP i MySQL koji nisu korišteni u ovom projektu.

Za verzioniranje projekta zaslužan je git-baziran sustav GitHub, dok je za dohvaćanje podataka zaslužan REST API.

## 1.2 Programsko rješenje

### 1.2.1 HTML

HTML je korišten kao temelj cijelog ovog projekta. Prije svega drugog, u *<head>* elementu definirane su poveznive na CSS dokumenta i D3.js biblioteku. U vidljivom dijelu web-stranice određenu su dvije sekcije: jedna za statistiku vezanu uz svijet, potom druga za statistiku vezanu uz pojedine države. Obje sekcije u sebi sadrže *<article>* elemente koji služe kao prostor za umetanje grafova. Kako bi bilo moguće razlikovati te elemente, svaki *<article>* ima posebnu klasu. Na kraju cijelokupne hijerarhije su paragrafi i upisni elementi.

Paragrafi ponajviše služe kao dodatan izvor informacije gdje se statistika daje izravno korisniku/klijentu. Kao i kod *<article>* elementata, i paragrafi su diferencirani korištenjem klasa. Paragrafi koji ne prenose određenu statistiku služe za grupiranje i opisivanje upisnih elemenata. U ovom projektu su korišteni gumbovi i jedan padajući izbornik kao upisni elementi. Definiranjem tih elemenata u samom HTML-u se ne postiže nikakva dinamična funkcionalnost – ta funkcionalnost će biti objašnjena tek u JavaScript dijelu rješenja. Na Slici 1.1. dan je primjer jedna HTML sekcije. Osim već navadenih elemenata, korišten je i *<h1>* element za prikazivanje naslova tog odjeljka. Na slici se također može vijdeti kako svaki element ima svoju određenu klasu. Detalji tih klasa će biti obrađeni u CSS dijelu rješenja. Još jedan bitan element je *<span>* element. Pomoću njega, prilikom dinamičkog mijenjanja teksta, moguće je samo mijenjati jedan dio teksta – u ovom slučaju samo brojke. *<span>* element također ima semantičku ulogu zbog istog razloga jer se pregledinku može dati do znjanja da se radi o naglasku jednog broja.



Slika 1.1. Primjer HTML strukture odjeljka za prikaz statistike po državi.

Na kraju web-stranice nalazi se zaglavlje sa njezinim kratkim opisom te autorskim podacima i podacima o korištenom API-ju. Na slici 1.2. je dan prikaz HTML strukture zaglavlja. Zaglavlje je podjeljeno na više djelova korištenjem *<div>* elementima. Svaki dio sadrži jedan od navedenih opisa. Zaglavlje samo po sebi ne igra veliku ulogu u vizualizaciji podatak, ali je ključan element svih web-stranica.



Slika 1.2. Prikaz HTML rješenja zaglavlja web-stranice.

### 1.2.2 JavaScript/D3.js

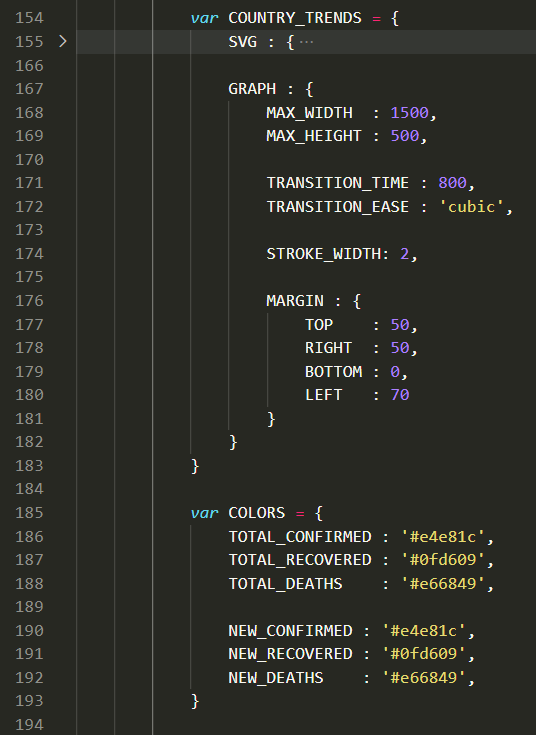
Cijelokupna JavaScript skripta je napisana unutar *<script>* HTML elementa te se zbog toga kod može činiti ne urednim. Ipak, kod je moguće podjeliti u tri osnova dijela:

1. Definiranje konstanti.
2. Dohvačanje podataka.
3. Stvaranje grafova.

#### 1.2.2.1 Konstante

U prvom dijelu se definiraju konstante koje pomažu ako dođe do naknadnih promjena tokom izvedbe projekta. Ovo se najbolje vidi u slučaju samih grafova kada je potrebno mijenjati jedan parametar na više mjesta, a taj parametar (poput visine npr.) se može često mijenjati. Slika 1.3. je isješak konstanti. Dan je primjer konstanti vezanih uz graf za pojedine države – sama konstanta je podjeljena na dva djela radi bolje preglednosti i organizacije podataka. Prvi dio definira visinu i dužinu *<svg>* elementa te njezine margine, dok drugi dio definira svojstva vezana uz sam graf i njegove tranzicije. Također se na isoj slici mogu vijdeti konstante vezane uz boje. Definiranjem boja osigurava se jednoliko korištenje boja u svim grafovima.

**// TODO: Add color explaination after you add NEW\_{} colors**



Slika 1.3. (155-183) JavaScript konstante vezane uz graf za pojedine države i (185-193) konstante vezane uz boje pojedinih parametara kod svih grafova.

#### 1.2.2.2 Dohvaćanje podataka

Drugi dio koda bavi se dohvaćanjem podataka iz vanjsog API-ja putem REST-a. To znači da je na određeni link (API) potrebno poslati ‘GET’ zahtjev kako bismo dobili željene podatke. U ovom projeku dolazi do spajanja na COVID-19 API. Svi podaci koji se dobiju natrag dolaze u JSON formatu. Prilikom prvog spajanja traže se općeniti podaci (engl. *summary*). Ti podaci sadrže sveukupno svjetsko stanje – koliko ima zaraženih, ozdravljenih i mrtvih – te koliko ima tih istih podataka, ali po državama. Ta dva skupa podataka služe za stvaranje prva dva grafa koji se nalaze u prvoj sekciji. Dio koda za prvo dohvaćanje se nalazi na slici 1.4. Iz nje se vidi kako je potrebno imati postban *XMLHttpRequest* objekt kako bi bilo moguće poslati zahtjev. Budući da je slanje zahtjeva asinkrona funkcija i da kod neće čekati dok ne dobije odgovor, ostatak koda nalazi se u funkciji *.onload* koja se poziva kada je odgovor poslan kako bi se program mogao slijedno izvršavati. Prije rada nad samim podacima provjerava se je li došlo do kakve greške u dohvaćanju. Funckcije *InsertGlobalSummary* i *InsertGlobalStats* služe za stvaranje već navedana prva dva grafa. Obje funkcije kao argumente primaju skup podataka nad kojim se graf stvara, klasu *<article>* elementa u kojem se nalaze te podaci za *<svg>* element i grafove. Svi od tih elementa, osim skupa podataka, su navedeni u dijelu sa konstantama.



Slika 1.4. Isječak koda za početno dohvaćanje podataka.

Prije opisivanja tih funkcija, dolazi do još dva dohvaćanja podataka. Drugo dohvaćanje je prikazano na slici 1.5. gdje se dohvaćanj *slugovi*, odnosno skraćeni zapisi imena, država, zbog toga što nemaju svi API-ji standardni zapis za države. Ovaj korak je nužan kako bi padajući izbornik, gdje se odvija treće dohvaćanje podataka, mogao funkcionirati. Padajući izbornik se inicijalizira pomoću funkcije *CountryPicker*.



Slika 1.5. Isječak koda za dohvaćanje imena/*slugova* država.

JavaScript pomoću D3.js-a dinamično kreira opcije u padajućem izborniku. Kao izvor podataka se predaju *slugovi* država te se svakom *slugu* stvara i pridjeljuje jedna opcija. Promjenom odabira pozova se funkcija *SelectChanged* prikazana na slici 1.6. Dohvaćanje podataka se vrši na isti način pomoću već kreiranog *XMLHttpRequest* objekta. Kada se podaci dohvate, stvara se graf za prikaz statistike COVID-19-a po državama.



Slika 1.6. Funckija *SelectChanged* koja se poziva nakon promjene odabira u padajućem izborniku.

#### 1.2.2.3 Iscrtavanje grafova

### 1.2.3 CSS

## 1.3 Vizualizacija podataka

### 1.3.1 Primjeri vizualizacije

# 2. KORIŠTENE IZVORI