

Integrált Informatikai Rendszerek Dolgozat

A koronavírus betegség, egy vírusos, légúti megbetegedés, amelyet a SARS-CoV-2 nevű koronavírus okoz. A járvány hivatalosan 2020 március 11.-én lett kihirdetve, a a Egészségügyi Világszervezet által. A betegség kezelésére sem hatásos gyógyszer, sem megelőzést lehetővé tevő védőoltás sem áll rendelkezésre. A vírus cseppfertőződésen keresztül terjed, Miután felfedezték a vírust Kína, Wuhan nevű városában, 2019 decemberében, néhány hónap múlva világszerte elterjedt.

A bemutató a COVID-19 világjárvány, romániai, helyzetéről hivatott tájékoztatni a lakosokat tehát a célközönség a román állampolgárok és bármilyen érdeklődő, akit foglalkoztat a romániai lakosság helyzete a koronavírussal szemben, mivel ahogy a vizualizációkon is látszik, majdnem 7%-os halálozási rátával jár a vírus, ami nem egy kis szám a 16 000 eddigi fertőzötthez. Semmi képp nem lehet kis számba venni a jelenlegi helyzetet, és fontos hogy tekintettel legyünk a koronavírussal szemben legsebezhetőbb korcsoportra, a 50 év feletti személyekre.

Véleményem szerint fontos hogy mindenki tájékoztatva legyen a helyzetről, mivel láhattuk hogy sokan figyelmen kívül hagyták a nemzetközi felhívásokat és szabályozásokat. A bemutató hivatott hangsúlyozni és informálni az embereket a helyzet súlyosságáról, mivel a látni lehet hogy

Az adataim az Nemzetközi Egészségügy Szervezet oldaláról szereztem, ahol naponta frissítik az információkat, így csak annyit kell tenni hogy betöltjük az adatokat jupyter lab-ba majd lefuttatjuk a kódot és updateli a következő napra szóló adatokat grafana-ban, ezzel is könnyítve az adatok a frissítését.

_county_code	_total_county	_total_healed	_total_dead	_county	date	lat	long
AB	309	22	17	Alba	2020-05-18	46.066667	23.583333
AR	589	1	74	Arad	2020-05-18	46.183333	21.316667
AG	259	0	20	Argeş	2020-05-18	44.85	24.866667
BC	430	0	49	Bacău	2020-05-18	46.566667	26.9
BH	504	0	19	Bihor	2020-05-18	47.066667	21.933333
BN	290	0	24	Bistriţa-Năsăud	2020-05-18	47.133333	24.483333
BT	634	0	42	Botoşani	2020-05-18	47.75	26.666667
BV	555	0	41	Braşov	2020-05-18	45.633333	25.583333
BR	20	0	0	Brăila	2020-05-18	45.266667	27.983333
B	1332	3	74	Bucureşti	2020-05-18	44.433333	26.1
BZ	119	0	0	Buzău	2020-05-18	45.15	26.833333
CS	103	4	6	Caras-Severin	2020-05-18	45.308333	21.889167
CJ	455	1	15	Cluj	2020-05-18	46.766667	23.6
CT	210	2	15	Constanţa	2020-05-18	44.183333	28.65
CV	214	0	9	Covasna	2020-05-18	45.866667	25.783333
CL	70	0	3	Călăraşi	2020-05-18	44.2	27.333333
DJ	150	0	3	Dolj	2020-05-18	44.316667	23.8
DB	254	6	10	Dâmboviţa	2020-05-18	44.933333	25.45
GL	494	0	81	Galaţi	2020-05-18	45.45	28.05
GR	187	0	4	Giurgiu	2020-05-18	43.883333	25.966667
GJ	99	1	7	Gorj	2020-05-18	45.05	23.283333
HR	68	0	0	Harghita	2020-05-18	46.35	25.8
HD	595	3	68	Hunedoara	2020-05-18	45.883333	22.9
IL	316	0	20	Ialomiţa	2020-05-18	44.566667	27.366667
IS	386	1	26	Iaşi	2020-05-18	47.166667	27.6
MM	68	1	3	Maramureş	2020-05-18	47.666667	23.583333
MH	94	0	12	Mehedinţi	2020-05-18	44.631944	22.656111

A következő oszlopokat tartalmazzák az adatok: Megye kód, Igazolt fertőzések száma, Felépültek száma, Halottak száma, a dátum oszlop szükséges volt a grafanával való kompatibilitás miatt, az utolsó két oszlop a megyék fővárosainak a koordinátáit tartalmazza.

A bemutató megvalósításához dockert használtam, azon belül, több, úgynevezett „container”-t, még hozzá, Jupyter notebook, InfluxDB és Grafana, példányok, amelyeknek külön-külön „container”-ük van, amelyeken keresztül tudnak kommunikálni egymással.

A Jupyter notebook-ot az adatok formázására és InfluxDB-be való elküldésére használtam, ugyancsak Jupyter notebook tette lehetővé hogy hozak létre egy saját adatbázist InfluxDB példányban, ahova fel tudtam tölteni táblákat. Miután elküldtem az adatokat, táblák formájában InfluxDB-be, szükséges volt specifikálni Grafanában hogy hol keresse az InfluxDB példányt és ezen belül melyik adatbázisban keresse az adatokat. Miután ez megtörtént egyszerűen ki kellett választani Grafanában hogy melyik táblát használjuk, melyik adatbázisból, ezzel készen állva, query-kkel tudtuk megmondani a vizualizációnak, miként jelenítse meg az adatokat.

Mivel Dockerben oldottam meg a bemutatót, amennyiben a példányokat külső tárolóra helyezük át, az alkalmazás skálázhatósága optimálissá válhat. Az elkészített rendszer a konténerizálásnak köszönhetően hatékonyan használja ki az erőforrásokat. Ez hatékonyabb egy virtuális géppel szemben, így skálázhatóbbá is válik ezzel a módszerrel a rendszer: könnyen átvihető más környezetekbe. A konténerizálás tehát a hordozhatóságot is növeli, hiszen a konténerek izoláltan futnak az operációs rendszertől, előre definiáltak a dependency és más kritériumok, ezáltal több, különböző verziójú rendszerrel rendelkező felhasználó használhatja ugyanazt az integrált informatikai rendszert a saját környezetében, hiba fellépése nélkül. A konténerizáláshoz a Docker-t használtam, hiszen ennek neve már egyenlő a konténerizáció fogalmával.

Bővítés szempontjából annyival lehetne fejleszteni az alkalmazást, hogy múltbeli adatokat összehasonlítom a jelen adatokkal és egy idősorban megjelenítek régebbi adatokat is, mivel a jelenlegi megoldás csak egy adott időpontra érvényes, ami a jelen. Viszont ezekhez az adatokhoz egy helyen, idősorban, nehéz hozzájutni, esetlegesen több idővel, össze tudnám gyűjteni ezeket és beiktatni a jelenlegi megoldásba.