|  |
| --- |
| V0.2 |
| Technisch Ontwerp Corona Dashboard |
| Project gemaakt voor REA college te Groningen |

|  |
| --- |
| Kjell Vos  22-10-2020 |

Inhoud

[1 Inleiding 2](#_Toc54872300)

[1.1 Achtergrond project 2](#_Toc54872301)

[1.2 Gebruikte documenten 2](#_Toc54872302)

[1.3 Standaards & Richtlijnen 2](#_Toc54872303)

[2 Systeemarchitectuur 3](#_Toc54872304)

[2.1 Architectuur 3](#_Toc54872305)

[2.2 Hardware 3](#_Toc54872306)

[2.3.1 Software 3](#_Toc54872307)

[2.4 Communicatie 3](#_Toc54872308)

[2.4.1 Gegevensopslag 3](#_Toc54872309)

[3 Gegevensmodel 4](#_Toc54872310)

[3.1 Data model van CSV 4](#_Toc54872311)

[4 Gebruikersinterface 5](#_Toc54872312)

[4.1 Scherm 5](#_Toc54872313)

[6 Processen 6](#_Toc54872314)

[6.1 Use Cases 6](#_Toc54872315)

## 1 Inleiding

Dit is het technisch ontwerp bij het corona dashboard project. Een corona dashboard is in dit geval een webpagina waar mensen makkelijk en overzichtelijk informatie geserveerd krijgen over het verloop van besmettingen van corona. Voor een duidelijk visueel beeld zou u kunnen kijken onder de kop 4.1 Scherm.

In dit document zal uitleg en toelichting gegeven worden die duidelijk maakt welke oplossingen zijn gekozen en waarom. Waarom er voor de gebruikte programmeertaal is gekozen.

### Achtergrond project

Dit project heb ik(Kjell Vos) gekregen als opdracht aan het REA College te Groningen. Dit project is alleen om te leren, Beide documenteren en het programmeren/visualiseren van data.

### Gebruikte documenten

Ik heb van het internet twee documenten gehaald.

Ik heb een JSON bestand gedownload die gebruikt wordt voor de vormen van de staten. Dit wordt gebruikt voor het tekenen van de ‘status’ kleur over de kaart.

De data van corona die ik heb gebruikt is te vinden hier onder dataset 1:

<https://data-flair.training/blogs/download-covid-19-dataset/>

### 1.3 Standaards & Richtlijnen

De documentatie zal bestaan uit een Plan van Aanpak en dit document het Technisch Ontwerp. De python code zal de PEP richtlijnen volgen, en goed met duidelijke commentaar beschreven worden.

Onder de python PEP richtlijnen zal ik hier een specifiek uitlichten die als extra belangrijk worden geacht voor duidelijke en leesbare code.

PEP 8:

<https://www.python.org/dev/peps/pep-0008/>

## 2 Systeemarchitectuur

Hier wordt de systeemarchitectuur en eventuele verschillende subsystemen nader behandeld.

### 2.1 Architectuur

Dit project is geprogrammeerd voor Python 3.9 en gebruikt 3 libraries, Folium(v0.11.0), Flask(v1.1.2) en pandas(v1.1.3).

Voor het programmeren is Pycharm 2020.2.1 gebruikt van JetBrains.

### 2.2 Hardware

Dit project is gemaakt op een Asus VivoBook Pro N580VD-FJ285T. Maar zou moeten werken op elk systeem waar een volledige python interpreter gedraaid kan worden met de juiste versie.

### Software

|  |  |
| --- | --- |
| Python | 3.9 |
| Flask | 1.1.2 |
| Folium | 0.11.0 |
| Pandas | 1.1.3 |

Er zijn meer libraries geïnstalleerd doordat Flask, Folium en Pandas afhankelijkheden hieraan hebben. Deze zijn niet genoteerd omdat deze automatisch bij gebruik van een goede package manager geïnstalleerd worden en verder in de code van dit project niet aangeroepen worden.

### 2.4 Communicatie

Dit project communiceert weinig met andere programma’s. De programma’s waar het project mee communiceert zijn een web browser om de data visualisatie te weergeven.

### Gegevensopslag

De enige gegevens die dit project gebruikt zijn een input CSV(Comma Seperated Values) bestand, Deze kan worden omgezet doormiddel van een script die bij het project is ingesloten naar een nieuwe CSV waar de data allemaal verwerkt is en gesorteerd en makkelijk te importeren en te verwerken is voor display van de visualisatie. Ik doe dit omdat anders ik elke keer als je de visualisatie laad in de browser de gehele data verwerkt moet worden. Ik vond het makkelijker om een script te hebben die de data groept en sorteert en verwerkt los van de webpagina.

## Gegevensmodel

Dit project werkt met CSV bestanden.

De dataset ‘covid-19-dataset-1.csv’ kun je met het ‘edit\_data.py’ script verwerken en de verwerkte data wordt opgeslagen in ‘data-edited.csv’ voor gebruik in folium/flask/pandas.

### 3.1 Data model van CSV

De originele ‘covid-19-dataset-1.csv’ heeft veel rijen en kolommen. Degene waar wij voor deze visualisatie geïnteresseerd in zijn zijn de velden ‘Confirmed’ en ‘Province\_State’ die aangeeft hoeveel corona besmettingen er zijn geweest. Deze data is nog niet per staat samengevoegd en gesorteerd. Er zijn dus meerdere rijen voor elke staat.

Alle kolommen van de ‘covid-19-dataset-1.csv’:

FIPS = Federal Information Processing Standard.

Admin2 = De stad/dorp waar de infectie is geregistreerd.

Province\_State = De Amerikaanse staat waar de infectie is vastgesteld.(Gebruikt)

Country\_Region = Welk land de infectie vastgesteld was.

Last\_Update = Laatste keer dat deze rij is geüpdatet.

Lat = Latitude coördinaat.

Long\_ = Longitude coördinaat.

Confirmed = Bevestigd aantal besmettingen.(Gebruikt)

Deaths = Hoeveelheid sterf gevallen.

Recovered = Hoeveelheid herstelde corona gevallen.

Active = Hoeveelheid actieve corona besmettingen

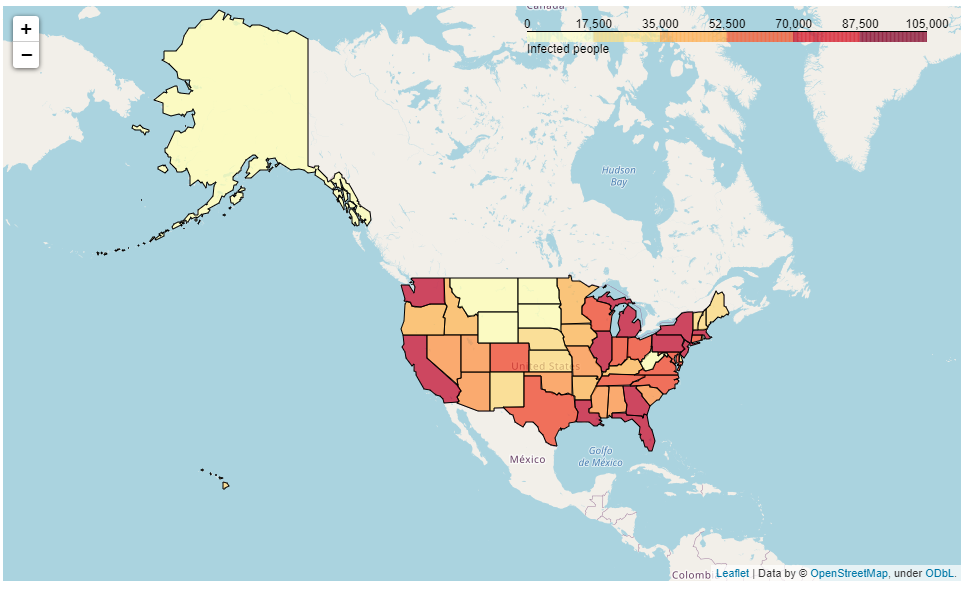
Combined\_Key = Admin2 + Province\_State + Country\_Region

Elke rij wordt verwerkt en er blijven 55 rijen over met alleen de staat code(AL=ALabama/TX=TeXas/NY=New York) en het aantal besmettingen wat opgeslagen wordt in een ander document, ‘data-edited.csv’.

## Gebruikersinterface

Er is niet veel aan gebruikersinterface aangezien het hele programma maar een webpage levert. Daarbij wordt het meeste van de UI gedaan door de libraries zodat er gefocust kan worden op data verwerking en visualisatie.

### 4.1 Scherm

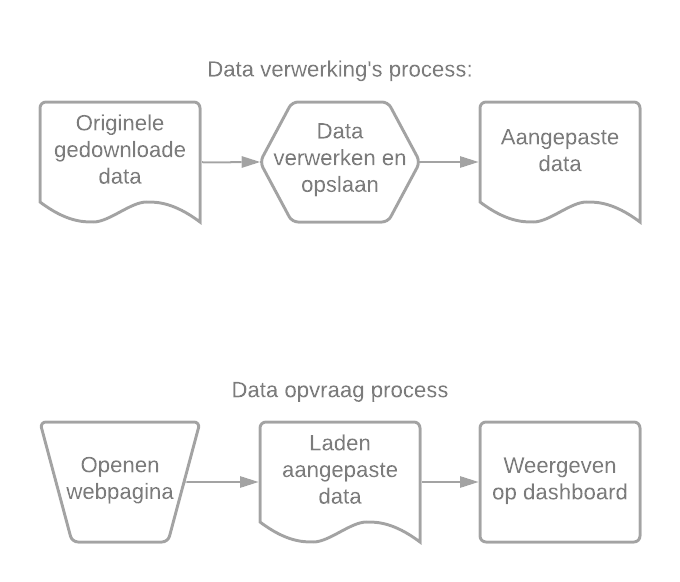


## 6 Processen

De processen van dit project zijn opgesplitst in 2 bestanden met elk een eigen functie.

Je hebt ‘edit\_data.py’ die gebruikt kan worden om de rouwe data die gedownload is aan te passen naar een makkelijker leesbaar format die wordt geschreven naar het bestand ‘data-edited.csv’.

En je hebt ‘main.py’ die het corona dashboard host, De data visualisatie opzet met de geladen data van ‘data-edited.csv’.



## 6.1 Use Cases

De gebruiker kan de pagina opzoeken in hun webbrowser om zo informatie te verkrijgen over de corona situatie.