**Documentation**

How to compile and/or run:

1. Install NET core SDK
2. Install MySql.Data package
3. Go to the database folder and run Schema.sql in MySqlWorkbench or run Schema.py to create the tables
4. Run AddData.py in the database folder to add the data into the database. All the datasets listed below should be in the datasets folder in the same directory as the script.
5. Run the server with visual studio or visual studio code, or run the exe in covidtracker/bin/release/covidtracker.exe

Datasets

**CSSE covid-19 daily reports**

<https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_daily_reports>

Vaccinations.csv FROM the Owid Covid dataset

<https://github.com/owid/covid-19-data/blob/master/public/data/vaccinations/vaccinations.csv>

WPP2019\_TotalPopulationBySex.csv From the UN. Total population – All variants

<https://population.un.org/wpp/Download/Standard/CSV/>

owid-covid-data.csv For some demographic data

<https://github.com/owid/covid-19-data/blob/master/public/data/owid-covid-data.csv>

Countries-Continents.csv to map countries to continents With some countries added manually

<https://github.com/dbouquin/IS_608/blob/master/NanosatDB_munging/Countries-Continents.csv>

Schema

Graphical user interface

Description automatically generated

Backend structure

**Controllers:**

DataController: Allows the retrieval of Covid related data from the database. All API Endpoints that start with /Data come from there. Uses the DataAccess service class.

UserController: Allows managers to log in and out. Uses the UserAccess service class

**Services:**

Data access: Retrieves Covid data from the database. Takes queries from the Model/Utils/Queries.sql file.

UserAccess: Checks if the user and password are in the database.

API

**/Data/LatestDate**

Get the latest date with covid reports in the database.

Example output: "2021-11-21T00:00:00"

**/Data/ByDate/Countries?Date=(date)&threshold=(int)**

Information on countries by date

(Threshold = number of days for new cases. Default is 14)

Example output:

/Data/ByDate/Countries?Date=2021-7-7&threshold=7

[

    {

        "countryID": 55,

        "countryName": "Afghanistan",

        "confirmedCases": 129021,

        "newCases": 21064,

        "deaths": 5415,

        "recovered": 77280,

        "population": 39943186,

        "vaccinated": 735213,

        "fullyVaccinated": 199250,

        "boosters": 0

    },…

]

**/Data/ByDate/Continents?Date=(date)&threshold=(int)**

Information on continents by date

(Threshold = number of days for new cases. Default is 14)

/Data/ByDate/Continents?Date=2021-7-7&threshold=7

Example output:

[

    {

        "continent": "Asia",

        "confirmedCases": 61570654,

        "newCases": 2337632,

        "deaths": 934933,

        "recovered": 58062058,

        "population": 4760951754,

        "vaccinated": 1187736763,

        "fullyVaccinated": 197678403,

        "boosters": 2432506

    },…

]

**/Data/ByDate/World?Date=(date)&threshold=(int)**

Information on the world by date

(Threshold = number of days for new cases. Default is 14)

Example output:

/Data/ByDate/World?Date=2021-7-7&threshold=7

{

    "confirmedCases": 223560578,

    "newCases": 8506989,

    "deaths": 4605343,

    "recovered": 1657483,

    "population": 7829114393,

    "vaccinated": 3281507893,

    "fullyVaccinated": 2324394973,

    "boosters": 23161770

}

**Data/Countries/{id}?start=(date)&end=(date)**

Information for a specific country

start and end are the date range for the information

Example output:

Data/Countries/114?start=2020-01-28&end=2021-05-05

"name": "Germany",

    "continent": "Europe",

      "pData": {

        "2020": {

            "population": 83783945,

            "density": 240.372,

            "povertyRate": **null**,

            "diabetesRate": 8.31,

            "medianAge": 46.6

        },

        "2021": {

            "population": 83720248,

            "density": 240.189,

            "povertyRate": **null**,

            "diabetesRate": 8.31,

            "medianAge": 46.6

        }

    },

    "diseaseData": [

        {

            "date": "2020-01-28",

            "confirmed": 4,

            "deaths": 0,

            "recovered": 0

        },…

],

"vaccineData": [

        {

            "date": "2020-12-15",

            "vaccinated": 28500,

            "fullyVaccinated": 28500,

            "boosters": **null**

        },…

]

**/Data/Continents/{name}?start=(date)&end=(date)**

Information for a specific continent

start and end are the date range for the information

Example output:

Data/Continents/South\_America?start=2020-01-23&end=2021-05-05

"populationsByYear": {

        "2020": 430457606,

        "2021": 434037528

    },

    "diseaseData": [

        {

            "date": "2020-01-23",

            "confirmed": 0,

            "deaths": 0,

            "recovered": 0

        },…

]

  "vaccineData": [

        {

            "date": "2020-12-31",

            "vaccinated": 8656,

            "fullyVaccinated": 0,

            "boosters": 0

        },…

]

}

**/Data/World?start=(date)&end=(date)**

Information for a specific continent

start and end are the date range for the information

Example output:

Data/World?start=2020-01-22&end=2021-05-05

"populationsByYear": {

        "2020": 7747900833,

        "2021": 7829114393

    },

    "diseaseData": [

        {

            "date": "2020-01-22",

            "confirmed": 556,

            "deaths": 17,

            "recovered": 30

        },…

]

 "vaccineData": [

        {

            "date": "2020-12-15",

            "vaccinated": 252625,

            "fullyVaccinated": 4287,

            "boosters": 0

        },…

]

**/Data/ByDate/PopulationData?year=(int)**

Gives the demographic data of each country by year.

Example output:

/Data/ByDate/PopulationData?year=2020

{

    "55": {

        "density": 59.627,

        "povertyRate": **null**,

        "diabetesRate": 9.59,

        "medianAge": 18.6

    },

    "98": {

        "density": 105.029,

        "povertyRate": 1.1,

        "diabetesRate": 10.08,

        "medianAge": 38

    },

…

**/Data/AddCovidReport**

Inserts more disease reports into the database

Input: CSV file delimited with commas. Contains the fields “country\_region”, “confirmed”, “deaths”, “recovered”. File name should be the date in the format mm-dd-yyyy.csv

Should be one of the files from here:

<https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_daily_reports>

Must be logged in to use this.

**/Data/AddCovidReport?date=(date)**

Inserts vaccine reports into the database

Date: only reports from at least this date should be inserted

Input: CSV file delimited with commas. Contains the fields “location”, “date”, “people\_vaccinated”, “people\_fully\_vaccinated”, “total\_boosters”

Should be the Vaccinations.csv file from here:

<https://github.com/owid/covid-19-data/tree/master/public/data/vaccinations>

Must be logged in to use this.

**/User/Login**

Logs the user in.

Example input:

{

    "Name": "Yosi",

    "Password": "password"

}

**/User/Logout**

Logs the user out

Complex Query examples:

---CountriesByDate---

-- Shows confirmed cases, deaths, recovered cases, population, number of people vaccinated for each country within the chosen date.

-- Also shows number of new cases over the amount of days specified in the @TH parameter (@TH = 14 means new cases over 14 days, etc)

-- vaccination data is taken from the closest date found before the input date

WITH

    VD AS (SELECT country\_id, MAX(date) as date FROM vaccine\_reports  WHERE date <= @DATE GROUP BY country\_id),

    VR AS (

            SELECT V.vaccinated, V.fully\_vaccinated, V.number\_of\_boosters, V.country\_id, V.date FROM vaccine\_reports AS V

            JOIN VD ON VD.country\_id = V.country\_id AND V.date = VD.date

        )

SELECT

    C.id,

    C.name,

    DR.confirmed,

    DR.confirmed - COALESCE(Prev.confirmed, 0) AS new\_cases,

    DR.recovered,

    DR.deaths,

    PR.population,

    COALESCE(VR.vaccinated, 0),

    COALESCE(VR.fully\_vaccinated, 0),

    COALESCE(VR.number\_of\_boosters, 0)

FROM disease\_reports AS DR

JOIN countries as C

    ON C.id = DR.country\_id AND DR.date = @DATE

JOIN population\_reports AS PR

    ON PR.year = YEAR(@DATE)

    AND PR.country\_id = DR.country\_id

LEFT JOIN disease\_reports AS Prev ON Prev.country\_id = DR.country\_id AND Prev.date = DATE\_SUB(@DATE, INTERVAL @TH DAY)

LEFT JOIN VR

    ON VR.country\_id = DR.country\_id

---ContinentsByDate---

-- Shows confirmed cases, deaths, recovered cases, population, number of people vaccinated for each continent within the chosen date.

-- Also shows number of new cases over the amount of days specified in the @TH parameter (@TH = 14 means new cases over 14 days, etc)

-- vaccination data is taken from the closest date found before the input date

WITH

    VD AS (SELECT country\_id, MAX(date) as date FROM vaccine\_reports  WHERE date <= @DATE GROUP BY country\_id),

    VR AS (

            SELECT V.vaccinated, V.fully\_vaccinated, V.number\_of\_boosters, V.country\_id, V.date FROM vaccine\_reports AS V

            JOIN VD ON VD.country\_id = V.country\_id AND V.date = VD.date

        )

SELECT

    C.continent,

    SUM(DR.confirmed),

    SUM(DR.confirmed) - COALESCE(SUM(Prev.confirmed), 0) AS new\_cases,

    SUM(COALESCE(DR.recovered, 0)),

    SUM(DR.deaths),

    SUM(PR.population),

    SUM(COALESCE(VR.vaccinated, 0)),

    SUM(COALESCE(VR.fully\_vaccinated, 0)),

    SUM(COALESCE(VR.number\_of\_boosters, 0))

FROM disease\_reports AS DR

JOIN countries as C

    ON C.id = DR.country\_id AND DR.date = @DATE

JOIN population\_reports AS PR

    ON PR.year = YEAR(@DATE)

    AND PR.country\_id = DR.country\_id

LEFT JOIN disease\_reports AS Prev ON Prev.country\_id = DR.country\_id AND Prev.date = DATE\_SUB(@DATE, INTERVAL @TH DAY)

LEFT JOIN VR

    ON VR.country\_id = DR.country\_id

GROUP BY C.continent

---WorldDiseaseData---

-- Gives the disease data of the world for each date between @START and @END

SELECT

    date,

    SUM(confirmed),

    SUM(deaths),

    SUM(COALESCE(recovered, 0))

FROM disease\_reports

GROUP BY date

HAVING date BETWEEN @START AND @END