CS209 Project Report

COVID19_DashBoard

11911109 张倚凡, 11910216 王标, 11911627 谈思序

Ⅰ. Brief introduction

In this project, we built a simple but useful covid19 data visulization web-app, which contains both frontend and backend. Components and techniques utilized are as follows:

- Frontend: Vue, webpack
- Backend: Springboot
- Database: PostgreSQL
- Data source: https://www.worldometers.info/coronavirus/ (Daily update), owid-covid-data.csv(Historcal data)
- Data process: Java

This *DashBoard* contains the following attributes for each country on each day:

- Total infected cases
- New infected cases
- Total deaths
- New deaths
- Total vaccinated
- New vaccinated

This *DashBoard* currently has the following functions:

1. Efficient **data strcture** to process the data and reliable **data storage**.
2. Data **table-view** display.
3. Support **4 methods of visulization**, including **world map**, line charts, dynamic charts, pie charts. With the well-designed backend and frontend, the visulization methods is scalable, which means user can cumstomize the charts with ease.
4. Automatically track the latest covid19 data using **web crawler**.
5. Support **search** and **sort** functions to display table-view.
6. Support **data export** to json and **image export**.
7. **Flexible parameters** can be set to display the data.

8. Support **animation** visulization.

File structure

Main file structure is shown as follows.

```
├─COVID19
│  ├─src
│  │  │  App.vue
│  │  │  index.ts
│  │  │  main.js
│  │  │
│  │  ├─assets
│  │  │      custom.geo.json
│  │  │      data.json
│  │  │      jquery-migrate-3.3.2.js
│  │  │      life-expectation-table.json
│  │  │      logo.png
│  │  │      medium.geo.json
│  │  │      world.json
│  │  │      world_en.json
│  │  │
│  │  ├─components
│  │  │      DynamicTrend.vue
│  │  │      FileSaver.js
│  │  │      LineChart.vue
│  │  │      PieChart.vue
│  │  │      Table.vue
│  │  │      WeeklyTrend.vue
│  │  │      world.json
│  │  │      WorldMap.vue
│  │  │
│  │  └─router
│  │          index.js
└─Springproject
    ├─src
    │  ├─main
    │  │  ├─java
    │  │  │  │  owid-covid-data.csv
    │  │  │  │
    │  │  │  ├─com
    │  │  │  │  └─example
    │  │  │  │      └─demo
    │  │  │  │          │  SpringprojectApplication.java
    │  │  │  │          │
    │  │  │  │          ├─config
    │  │  │  │          │      CorsConfig.java
    │  │  │  │          │
    │  │  │  │          ├─controller
    │  │  │  │          ├─entity
    │  │  │  │          └─web
    │  │  │  │                  dataProess.java
    │  │  │  │                  HelloTest.java
    │  │  │  │
    │  │  │  ├─config
    │  │  │  │      Config.java
    │  │  │  │
    │  │  │  ├─data
    │  │  │  │      Continent.java
    │  │  │  │      Country.java
```

```
    |   |   |   |      CountryCase.java
    |   |   |   |      CountryData.java
    |   |   |   |      DataReadEncapsulation.java
    |   |   |   |      DataToJson.java
    |   |   |   |      DataUtil.java
    |   |   |   |      DynamicData.java
    |   |   |   |      InfoPiece.java
    |   |   |   |
    |   |   |   └─database
    |   |   |           Crawler.java
    |   |   |           DatabaseAccess.java
    |   |   |           SQLDataSource.java
    |   |   |
    |   |   └─resources
    |   |       |   application.properties
    |   |       |   config.properties
    |   |       |
    |   |       ├─static
    |   |       └─templates
    |   └─test
    |       └─java
    |           └─com
    |               └─example
    |                   └─demo
    |                           SpringprojectApplicationTests.java
```

Ⅱ Class Methods and Fields

Data structure

- **continent**

  ```
  String name;
  long value;
  ```

  pie charts need continent dataType to express the total cases of every continent.

- Country

  ```
  private String CountryCode;
  private String CountryName;
  private String Continent;
  public ArrayList<InfoPiece> infoList;
  private int total_cases;
  private int total_deaths;
  private int total_Vacs;
  ```

  Country dataTpye is to store all **infoPieces** of this country and relevant information.

- countryCase

```
String name;
Long value;
String Continent;
```

Map chart needs countryCase dataType to represent the total cases of every country.

- countryData

```
ArrayList<String> date;
ArrayList<Long>  newDeaths;
ArrayList<Long>  newVACs;
ArrayList<Long>  newCases;
```

Line chart needs countryData dataTpye to express from the data on, in the next six day, what newDeaths, newVACs, newCases are.

- dynamicData

```
String date;
String  country;
long cases;
```

dynamic chart needs dynamicData to express the total cases of the country in this date.

- InfoPiece

```
public  String CountryCode;
public  String CountryName;
public  String Continent;
public  Date date;
public  Long newCases;
public  Long totCases;
public  Long newDeaths;
public  Long totDeaths;
public  Long newVACs;
public  Long totVACs;
```

InfoPiece dataTpye is to represent every line of the crawler data.

Data process

The whole process is as follows.

- accept request from the webpage
  dataProcess

```
    @GetMapping("/continent")
    public String continent(){}

    @GetMapping("/country")
    public String country(@RequestParam String countryName,String date) throws ParseExcept

    @GetMapping("/map")
    public String map(){}

    @GetMapping("/dynamic")
    public String dynamic(){}

    @GetMapping("/table")
    public String table(@RequestParam String date, String group, String order) throws Pars
```

- select the corresponding data according to the passed parameters.

  DataUtil

```
public static String mapChartData(ArrayList<InfoPiece> records){}

public static String animaData(ArrayList<InfoPiece> records){}

public static String pieChartData(ArrayList<InfoPiece> records){}

public static String tableData(ArrayList<InfoPiece> records, String group, String order,

public static String tableData(ArrayList<InfoPiece> records, String group, String order,
```

- encapsulate the data and return it to the request as Json

  DataToJson

```
public static String tableData(ArrayList<InfoPiece> records, String group, String order,

public static String countryDataGet(Country country, Date date) throws ParseException {}

public static String dynamicDataGet(ArrayList<InfoPiece> records) {}

public static String mapDataGet(CountryCase[] country_cases) {}

public static String tableDataGet(ArrayList<InfoPiece> records) {}
```

Database

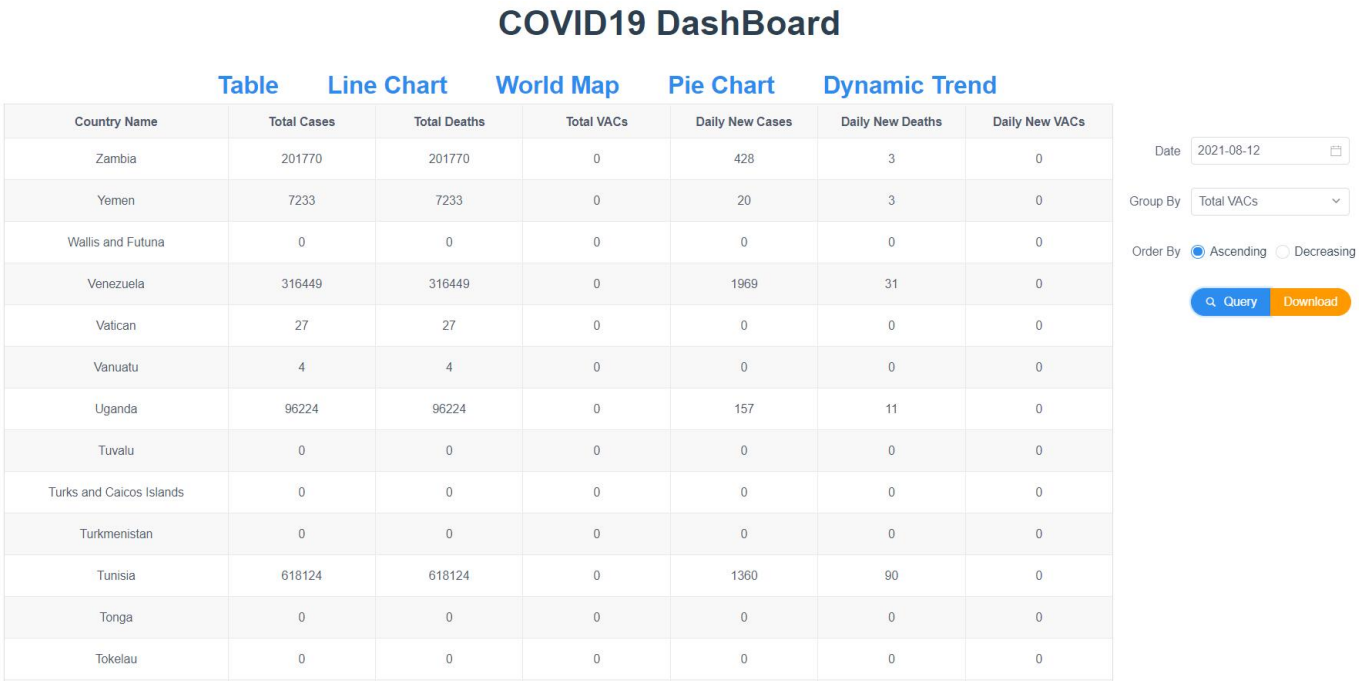
DatabaseAccess

This methods are used to insert data into database and retrieve data, as their name indicate.

```
public static void insertPiece(InfoPiece piece){}
public static ArrayList<InfoPiece> getRecords(PreparedStatement pstm){}
public static ArrayList<InfoPiece> getAllRecords(){}
```

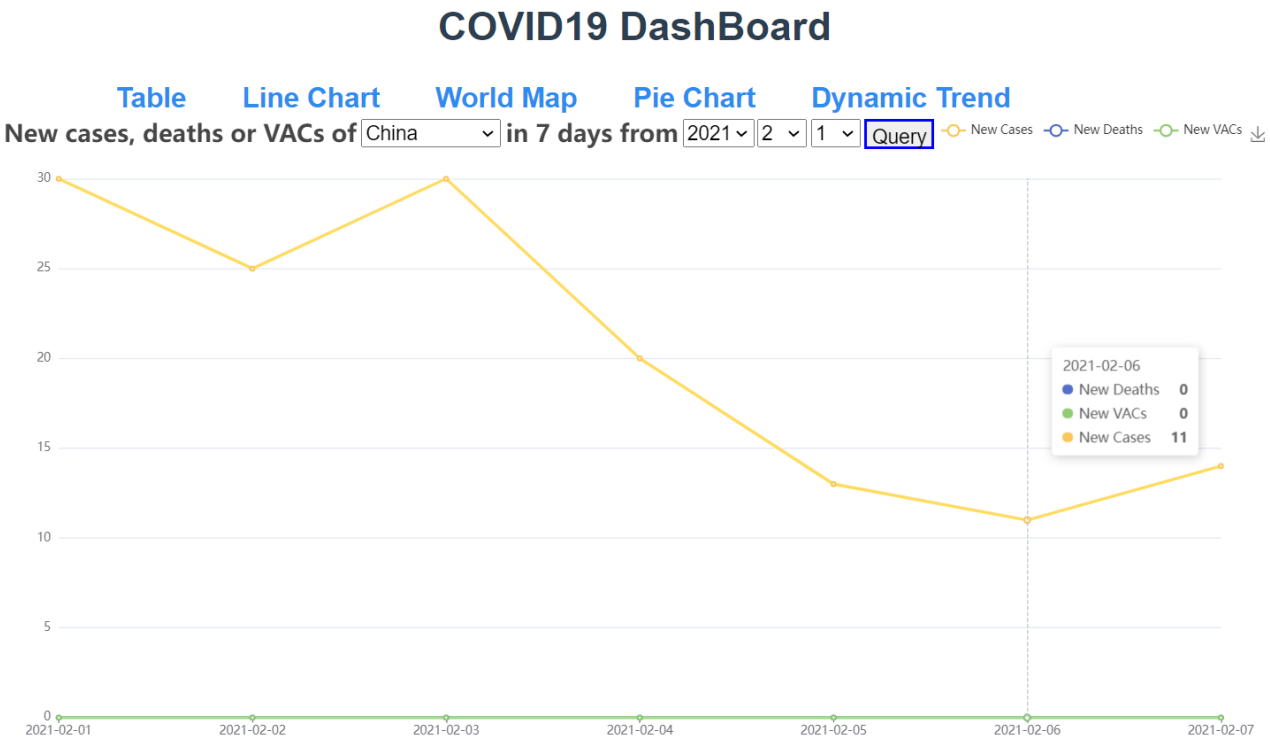
Ⅲ. Demonstration

Table-view with search and sort function



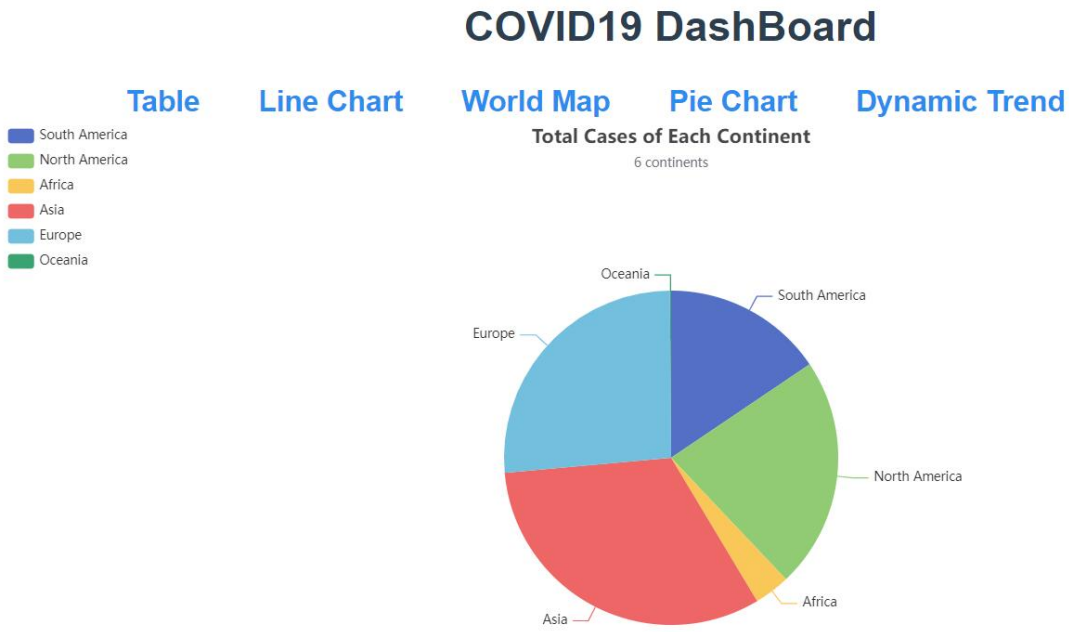
As shown, search and sort can be invoked using the floating selector on the right side.

Line-chart



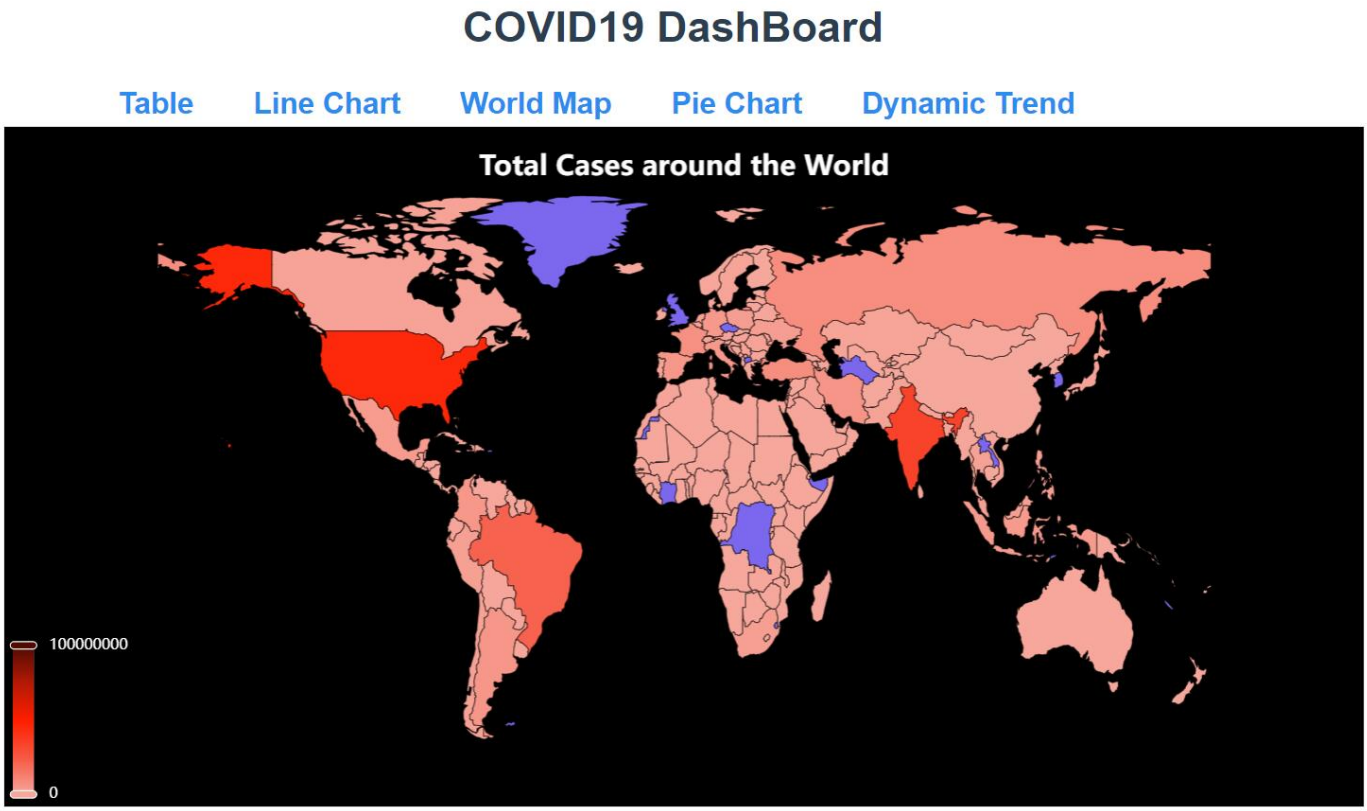
Line charts shows the new cases, deaths or VAcs of a selected Country in the last 7 days from ceitain day.

Pie-chart



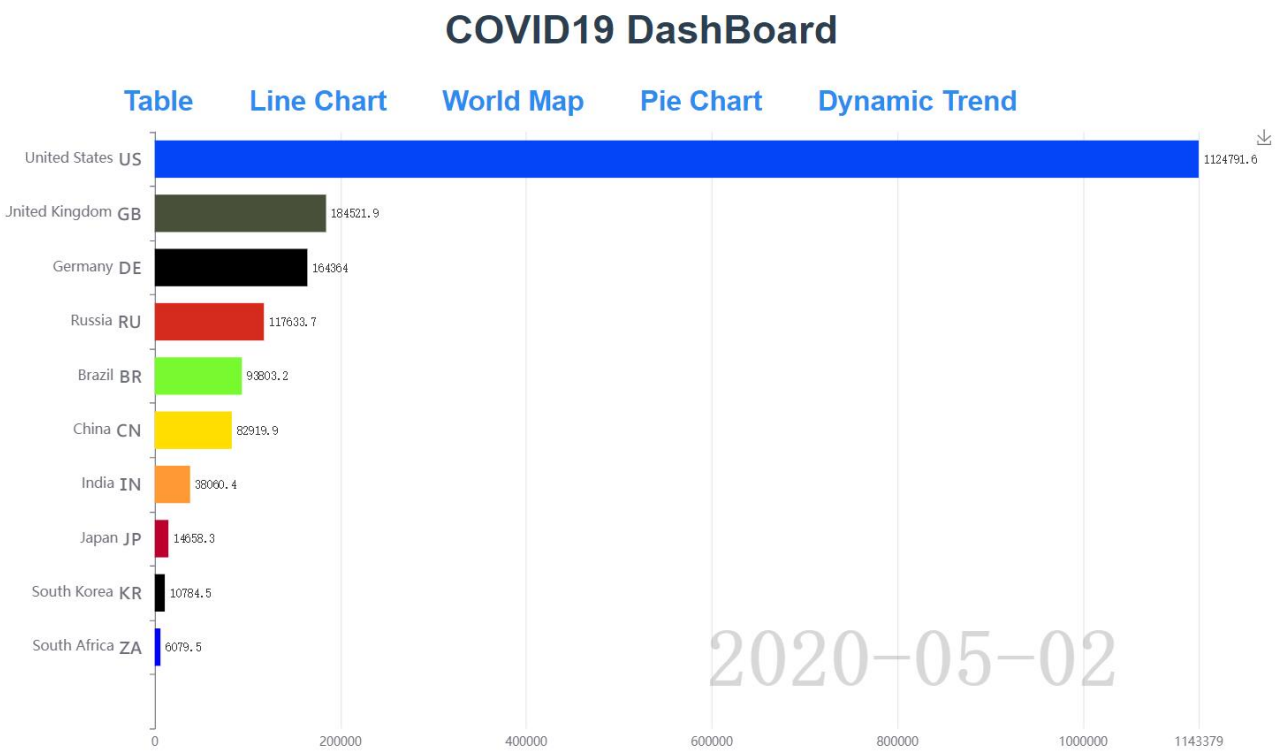
Pie charts presents the COVID19 situation among all continents.

World Map



Map shows total cases of a certain country all around the world.

Trend Animation



This dynamic animation figure shows the total cases of 10 countries.

Realtime database

Output | Covid19DashBoard.public.daily_data

1-500 of 501+ | Tx: Auto | DDL | CSV

	country	continent	day	new_case	tot_case	new_death	tot_death	new_vac	tot_vac
1	Afghanistan	Asia	2021-12-20	36	157787	2	157787	82	145234
2	Afghanistan	Asia	2021-12-19	1	157745	1	7333	34	145137
3	Afghanistan	Asia	2021-12-18	19	157744	0	7332	0	145103
4	Albania	Europe	2021-12-20	0	205549	0	205549	0	197055
5	Albania	Europe	2021-12-19	0	205224	0	3158	0	196700
6	Albania	Europe	2021-12-18	296	205224	2	3158	402	196700
7	Algeria	Africa	2021-12-20	0	214330	0	214330	0	147448
8	Algeria	Africa	2021-12-19	0	214044	0	6175	0	147263
9	Algeria	Africa	2021-12-18	299	214044	4	6175	194	147263
10	Andorra	Europe	2021-12-20	0	20549	0	20549	0	18285
11	Andorra	Europe	2021-12-19	0	20549	0	134	0	18285
12	Andorra	Europe	2021-12-18	0	20549	0	134	0	18285
13	Angola	Africa	2021-12-20	0	65868	0	65868	0	63743
14	Angola	Africa	2021-12-19	0	65760	0	1738	0	63691
15	Angola	Africa	2021-12-18	112	65760	1	1738	56	63691
16	Anguilla	North America	2021-12-20	0	1592	0	1592	0	1521
17	Anguilla	North America	2021-12-19	0	1592	0	4	0	1521
18	Anguilla	North America	2021-12-18	0	1592	0	4	0	1521
19	Antigua and Barbuda	North America	2021-12-20	0	4198	0	4198	0	4039

With web crawler, the most up-to-date data can be retrieved. This figure presents part of data scraped from the internet.

IV Acknowledgement

Thanks all the team members, they stayed up late for a whole weekend to finish this fancy project. Thanks Meeting room 804B in College of Engineering, where we combated the codes.