By: Geetha Madhuri Chittuluri

CS510-Front End Web Technologies

Project – COVID-19 Tracker Application

Public GitHub Repo: https://github.com/geetham04/FrontEnd-CoronaVirus-Dashboard-Application

Project URL: https://geetham04.github.io/FrontEnd-CoronaVirus-Dashboard-Application/

Overview:

About the application:

To spread the awareness of the current situation – Spread of Corona Virus, built the COVID-19 Tracker, a dashboard application, where one can visualize the data about affected and recovered cases throughout the entire world.

This dashboard application has got the following features – Tracking global wise Infected, Recovered and Death cases data information, Tracking the data by choosing a specific country from country picker and visualizing it in the form of a bar chart, Animation on counting number of Infected, Recovered and Death cases.

High level Design:

a. This is a single page web application. On load of the application, display of current time with a welcome greeting to this application and then a title in the image tag would be seen.

b. And then cases data all over the world till date will be shown in cards as infected, recovered and deaths. Below that a country picker is shown with a list of countries coming from API along with global.

c. By default, global will be shown as selected, where it displays daily cases growth in line chart and daily infected cases timeline in bar chart.

d. When clicked on a specific country from the country picker, display of that country specific cases data shown in various charts like bar, pie, doughnut and polar.

e. And then a footer section would be seen with a copyright information.

f. When you are at the bottom of the page, a scroll up icon is shown, if in case any one wants go to the top section of the page instead of scrolling, a click would take there.

Features:

1. Designed cards to show the infected, recovered and death cases data information till date.

2. Tracking global wise Infected, Recovered and Death cases data information and visualizing them through a line chart.

3. Tracking the data by choosing a specific country from country picker and visualizing it in the form of various charts.

4. Animation on counting numbers of Infected, Recovered and Death cases.

5. Displaying current time with a welcome greeting, added scrollup button

Stack:

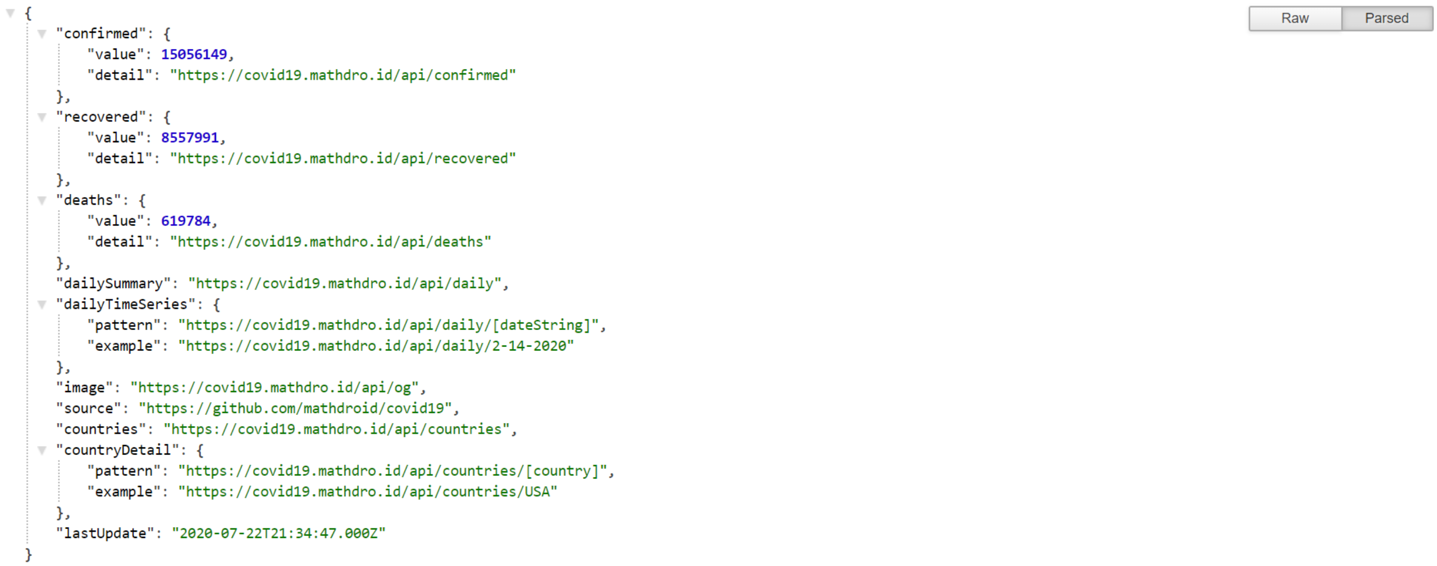
Languages/Frameworks – CSS, JavaScript, React

Charting Libraries – ChartJS (JavaScript charting library)

API:

https://covid19.mathdro.id/api

Few words about the API – To view in a formatted way, I have installed JSON formatter, a chrome extension, which parsed the data as shown below –



```
{
    "confirmed": {
        "value": 15056149,
        "detail": "https://covid19.mathdro.id/api/confirmed"
    },
    "recovered": {
        "value": 8557991,
        "detail": "https://covid19.mathdro.id/api/recovered"
    },
    "deaths": {
        "value": 619784,
        "detail": "https://covid19.mathdro.id/api/deaths"
    },
    "dailySummary": "https://covid19.mathdro.id/api/daily",
    "dailyTimeSeries": {
        "pattern": "https://covid19.mathdro.id/api/daily/[dateString]",
        "example": "https://covid19.mathdro.id/api/daily/2-14-2020"
    },
    "image": "https://covid19.mathdro.id/api/og",
    "source": "https://github.com/mathdroid/covid19",
    "countries": "https://covid19.mathdro.id/api/countries",
    "countryDetail": {
        "pattern": "https://covid19.mathdro.id/api/countries/[country]",
        "example": "https://covid19.mathdro.id/api/countries/USA"
    },
    "lastUpdate": "2020-07-22T21:34:47.000Z"
}
```

From the above API data, I used, the number of confirmed, recovered and deaths cases, dailySummary – daily data, source, countries, countryDetail – for a specific country – get the current data, along with the time of last update.

Technical Details of What I built:

Designed using React JS, CSS: In this React application, used one single class-based component, and rest of the components are functional components with hooks. Organized as a folder structure for all the components and the API fetching itself, everything is styled, up-to-date by following best practices.

First-of-all, initialized an empty react application using npx create-react-app ./

Main Files – index.js, App.js

Before starting the react application, installed the below necessary dependencies –

- Axios – with the help of axios, made get request to API
- React-chartjs-2 – one of the better libraries for making charts
- React-countup – used for the animation while counting numbers
- @material-ui/core - this helps with mobile responsiveness and also going to help without having to do - so much css. Just going to use materialUI components

using npm install –save axios react-chartjs-2 react-countup @material-ui/core classnames

npm start - starts react application on port localhost:3000

It’s really important to use modules in react, if we don’t use css modules, then all the styles would be applied to all components, which leads to interference and unpredictable code.

Index.js file inside components folder is going to take all the components cards, chart and countrypicker, Timer, Header and Footer and is going to export them from the components folder.

App is class-based component, whereas others are functional with hooks.

App is the parent component of Cards, Chart, Header, Footer and Countrypicker

Next step is creating API calls:

API folder is created – Inside that index.js file created – Inside this, made some functions that are going to fetch some data that needed.

Now, walking through the implementation steps in the flow shown in dashboard application-

1. Cards are always being shown and the data is being fetched right from the APP.js

2. Countrypicker with global and countries displayed in a dropdown –

In components folder – countrypicker.jsx – I have a countrypicker component and it works using the formcontrol and nativeselect from materialUI, there I am fetching the countries from the API, it’s just an array of 189 countries, just looping over them and then able to choose from all the countries. Once the country chosen, it goes back to app.js to dealCountryChange – there I get the country as a parameter and making the request one more time to the fetchData API. This time, with the fetchedData, it also changes the country.

3. With Global selected from countrypicker dropdown, displays daily cases growth in a line chart and daily infected cases timeline in a bar chart –

The data is being fetched right from Chart.jsx i.e., fetchDailyData, which is completely different data coming from API, it’s historic daily data, looping over this data and displayed it in line chart which is using the react-chartjs-2. And then, in the same way, daily infected cases timeline data is being fetched from fetchDailyDataPerDay from API, looping over this data and displayed it in bar chart.

4. With a specific country selected from countrypicker dropdown, displays country specific cases data (Infected, Recovered, Deaths) in various charts –

When chosen a specific country, in API, don’t have the array of all the different days of the data, just have the current data of confirmed, recovered and deaths cases i.e., one number for each case coming from the API.

5. Animation on counting numbers of Infected, Recovered and Death cases –

With the react countup component – implemented the animation on counting the number of cases. Getting all the data, coming from props from the APP.js which is using the API and then rendering 3 different cards showing the current data in this countup component.

6. Finally, index.js in api folder – have all the APIs – fetching the data, fetching the daily cumulative data, fetching the daily data per day count and then fetching countries.

7. Used async await, used proper react folder structure and the files.

8. Exported all the components from only one file i.e., index.js under components folder.

9. In App.js, displaying the header with current time, greeting message, image, cards, countrypicker, chart and footer.

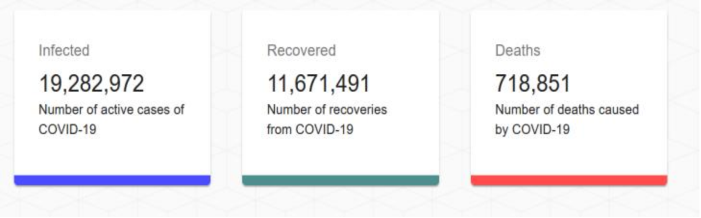
Issues Faced:

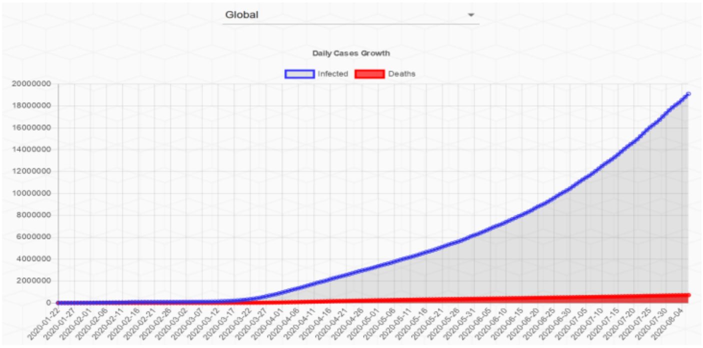
Page rendering issues –

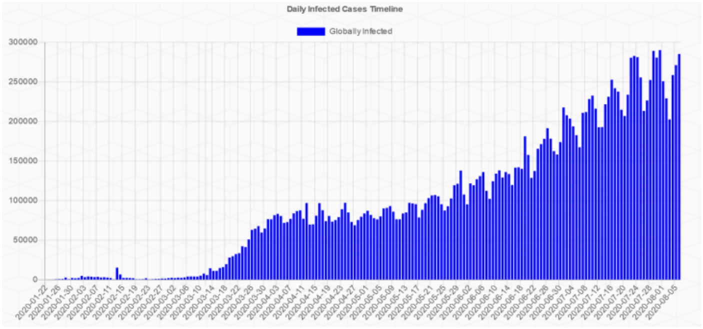
Changing a specific country to global in country picker – error (cannot get API/Countries/global)

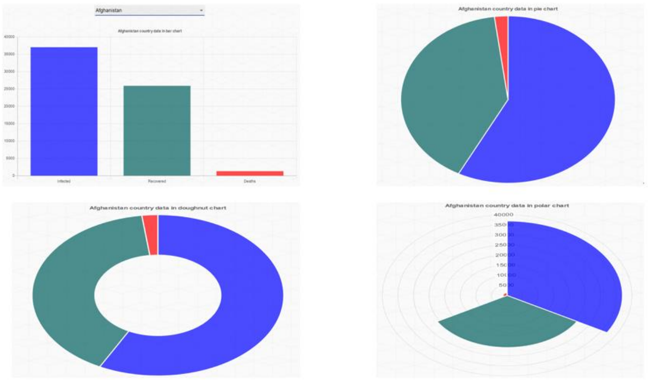
Solved: In countrypicker jsx file, I set the value for global option as global where I have to set an empty string instead as there is no country chosen. This fixed the problem

Screenshots:

Cards showing the infected, recovered and death cases data information till date	 Infected 19,282,972 Number of active cases of COVID-19 Recovered 11,671,491 Number of recoveries from COVID-19 Deaths 718,851 Number of deaths caused by COVID-19

<table>
<tr>
<td>Visualizing daily cumulative Global wise infected and death cases data in a line chart</td>
<td>


</td>
</tr>
<tr>
<td>Visualizing daily infected cases globally through the bar chart</td>
<td>


</td>
</tr>
<tr>
<td>Visualizing country specific infected, recovered and death cases data through various charts</td>
<td>


</td>
</tr>
</table>