1. Create react app folder
2. Installing required libraries for react

npm install --save axios

react-chartjs-2 react countup classnames

react countup classnames

Combining all these, we write in one line like

-> npm install --save axios react-chartjs-2 react countup classnames

3. Creating folders and files in it

4. We have made .module.css file in each folder, it is used to apply css only files in that folder. Like card.module.css apply css in only card.jx, not on whole application

5. make app.module.css to apply some css on only app.js, we import it In variable styles, so whenever we need to call some property from it we use className={styles.propertyname} in require tag

import styles from './App.module.css';

We do this to avoid interference of any of the css file.

6. Creating api folder and create api.js file and in that, we perform fetching in asynchronous function so that our app will not crash while fetching the data.

import axios from 'axios';

const url = "https://covid19.mathdro.id/api";

export const fetchData = async () =>{

    try{

        const response = await axios.get(url);

        return response

    }

    catch(error)

    {

    }

}

Here axios is used to request data from the server

7.fetching api using useEffect and use asynchronous function in it to avoid getting app crashed, and pass that globalData in different componets which can be used using props.data.confirmed etc

 const [globalData, setData] = useState({});

  useEffect(()=>{

    async function fetchGlobalData(){

      const response = await fetch("https://covid19.mathdro.id/api")

      //console.log(response);

      const dataFromAPI = await response.json();

      //console.log(dataFromAPI);

      setData(dataFromAPI)

    }

    fetchGlobalData();

  })

  return (

    <div className={styles.container}>

      <h1>Hello</h1>

      <Cards data={globalData}/>

      <Charts/>

      <CountryPicker/>

    </div>

  );

8. Now install material ui libraries to design our app

9. Create one main grid of spacing 3 means 3 contents can lies in it, so we created 3 child grids in it each for one card separately. And call data using destructuring as described in below point

9. this is the destructuring of the data:

{data: { confirmed, recovered, deaths, lastUpdate }

Here confirmed is like data.confimed, if look at api, the value lies in command data.confirmed.value

So we destructure data.confirmed as confirmed. Now to get confirmed cases value, instead using whole complete data.confirmed.value, we can use simple confirmed.value as we destructure data.confirmed as comfired. Same with other properties.

10. Countup is used to show numbers couting up starting from 0 to given value separated by , using separator

import CountUp from 'countup';//to show numbers counting up from starting to ending position separated by comma

<CountUp start={0} end={confirmed.value} duration={2.5} separator=","/>

11. Used to show dates in human readable form

new Date{lastUpdate.toDareString}

12. To use multiple classes in css tag, we use

yarn add classnames

import cx from 'classnames';

<Grid item component={Card} xs={12} md={3} className={cx(styles.card, styles.infected)}>

By using cs in classname, we can call multiple css classes in className

13. working in cards.module.css

14. To make chart and visualize the daily data, we need to fetch daily data from api, so we created new async function to fetch daily data in api.js in folder api

export const fetchDailyData = async () => {

    try {

        const { data } = await axios.get('https://covid19.mathdro.id/api/daily');

        const modifiedData = data.map((dailyData)=> ({

            confirmed: dailyData.confirmed.total,

            deaths: dailyData.deaths.total,

            date: dailyData.reportDate,

        }));

        return modifiedData;

    } catch (error) {

    }

}

Here we map data is an object so we map it to get daily confirmed and deaths cases and date and store in const modifiedData and return it for use

Now in order to display data in charts, we call it i

import { Line, Bar } from 'react-chartjs-2';

const Charts = () => {

    const [dailyData, setDailyData] = useState([]);

    useEffect (()=> {

        const fetchAPI = async () => {

            setDailyData(await fetchDailyData());

            console.log("daily Data=", dailyData);

        }

//importing daily data from api folder

import { fetchDailyData } from '../../api/api';

import { Line, Bar } from 'react-chartjs-2';

const Charts = () => {

    const [dailyData, setDailyData] = useState([]);

    useEffect (()=> {

        const fetchAPI = async () => {

            setDailyData(await fetchDailyData());

            console.log("daily Data=", dailyData);

        }

        fetchAPI();

    }, []);

    const lineChart = (

        dailyData.length

        ? (

            <Line

            data={{

                labels: dailyData.map(({date})=>date),

                datasets: [{

                    data: dailyData.map(({ confirmed })=>confirmed),

                    label: 'Infected',

                    borderColor: '#3333ff',

                    fill: true,

                }, {

                    data: dailyData.map(({ deaths })=>deaths),

                    label: 'Deaths',

                    borderColor: 'red',

                    backgroundColor: 'rgba(255, 0, 0, 0.5)',

                    fill: true,

                }],

            }}

        />):null

    );

    return (

        <div className={styles.container}>

            {lineChart}

        </div>

    )

}

export default Charts;

WE import fetchDailyData function from api. Create useEffect function to update our app chart in run time by updating value using function provide by the useState. setDailyData will update the value everytime.

Now we are making chart and saving it in const line chart. If dailyData has some value then the next line will run other wise it returns null.

We call line chart tag as <Line/>

Our labels our date, we our ploting infected and deaths. We pass two datasets, one for the infected and one for the deaths as shown below

data={{

                labels: dailyData.map(({date})=>date),

                datasets: [{

                    data: dailyData.map(({ confirmed })=>confirmed),

                    label: 'Infected',

                    borderColor: '#3333ff',

                    fill: true,

                }, {

                    data: dailyData.map(({ deaths })=>deaths),

                    label: 'Deaths',

                    borderColor: 'red',

                    backgroundColor: 'rgba(255, 0, 0, 0.5)',

                    fill: true,

                }],

            }}