ABSTRACT

The novel coronavirus 2019 (COVID-2019), which first appeared in Wuhan city of China in December 2019, spread rapidly around the world and became a pandemic. It has caused a devastating effect on both daily lives, public health, and the global economy. It is critical to detect the positive cases as early as possible so as to prevent the further spread of this epidemic and to quickly treat affected patients. The need for auxiliary diagnostic tools has increased as there are no accurate automated toolkits available. Recent findings obtained using radiology imaging techniques suggest that such images contain salient information about the COVID-19 virus.

Three months since the World Health Organization (WHO) declared the outbreak of COVID-19 as a pandemic, the health crisis has wreaked havoc on people's lives and livelihoods across the globe. Can state responsibility be apportioned for the pandemic, under the current international legal system? What would the elements of such responsibility be? This brief explores the concept of "state responsibility" under public international law and examines whether China—ground-zero of the pandemic—can be made legally responsible. The brief studies practical cases to assess how principles of international law have previously been applied with respect to state responsibility.

ACKNOWLEDGEMENT

I convey my sincere thanks to Mr. Ranjit Kumar (Assistant Professor, Department

of Information and Technology, Galgotias College of Engineering and Technology) for continuous inspiring help and encouragement throughout the

project and also for giving necessary advice and guidance and arranging all

facilities to make my learning easier. I choose this moment to acknowledge his

contribution gratefully.

I am also thankful to the Head of Department of Information Technology, Sanjeev

Kumar Singh for constantly supporting me in my work. His availability and creative suggestions were above and beyond what I expected. His patience,

understanding, encouragement and personal guidance have proven invaluable in

the preparation of this work. I appreciate his detailed discussions for deepening my

understanding of my chosen field.

Last but not the least I am grateful to all my friends who supported and encouraged

me to complete my work. They were my strong pillars of support who never failed to present their immaculate suggestions and thus helped me in giving shape to this

project as it is today.

Finally, I would like to thank the almighty for providing me the necessary health

and strength and also my family - my parents. I am indeed blessed to have such

parents who gave me the limitless freedom to explore this vast field.

Place: Greater Noida

Yours Sincerely,

Date: 11-1-21

Dileep Pal Mayank Tiwari Md. Shamsuzaman

TABLE OF CONTENT

1. Acknowledgement	3
2. Introduction.	4
3. Purpose of the project.	5
4 . Objective of the product	6
5. Features and Application.	8
6. Material-UI.	9
7. Chart.js	10
8. API	11
9. Implementation	13
9. Result analysis	
10. Conclusion, Limitation and Future scope	28
11. References.	30

INTRODUCTION

The growing activation in the dissemination of digital technologies in the economy and society is manifested in the expansion of automation of processes associated with the management of the organization and extends to all subsystems, including human resource management. The real opportunity on the Internet is not just doing what you have always done cheaper and faster, but instead, the real opportunity is to rethink at a fundamental level the business models that you employ on this new platform, both in terms of what kind of value you can deliver to your customers and also the kinds of relationships that you can build with customers.

• PURPOSE OF THE PROJECT

This project is to design and build a general awareness about the severity of the pandemic. The implementation of this project is to provide a large all the boundaries at an affordable price. The major's advantages of the digital awareness are:

- Global reach a website allows you to find current situation around the world.
- Lower cost a properly planned and well-targeted website campaign can reach the right customers at a much lower cost than traditional marketing methods.
- Trackable, measurable results you can use graphs to realise the current trend of increase in active cases of COVID-19.
- Personalization Visitors can view the number of cases in their own country.
- Openness by getting involved with social media and managing it carefully, you can build customer loyalty and create a reputation for being easy to engage with.
- Social awareness A website lets you create engaging campaigns using valuable content. This content (images, videos, articles) can increase social awareness being passed from user to user and becoming viral.

• OBJECTIVE OF THE PRODUCT

The main objective of this project is to develop a general awareness about the severity of the pandemic. This is carried out by the following objectives:

- Developing an interactive and user-friendly website.
- Providing real time updates
- Content writing
- Graphs

FEATURES AND APPLICATIONS

REACT(web framework)

React (also known as React.js or ReactJS) is an open-source, front end, JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing. React Route is an example of such a library.

Basic usage

The following is a rudimentary example of React usage in HTML with JSX and JavaScript.

```
<div id="myReactApp"></div>
<script type="text/babel">
function Greeter(props) {
  return <h1>{props.greeting}</h1>
}
  var App = <Greeter greeting="Hello World!" />;
  ReactDOM.render(App, document.getElementById('myReactApp'));
</script>
```

The Greeter function is a React component that accepts a property greeting. The variable App is an instance of the Greeter component where the greeting property is set to 'Hello World!'. The ReactDOM.render method then renders our Greeter component inside the DOM element with id myReactApp.

```
When displayed in a web browser the result will be <div id="myReactApp"> <h1>Hello World!</h1> </div>
```

MATERIAL-UI

Material-UI provides an optional CssBaseline component. It fixes some inconsistencies across browsers and devices while providing slightly more opinionated resets to common HTML elements.

Material-UI components work in isolation. They are self-supporting, and will only inject the styles they need to display. They don't rely on any global style-sheets such as normalize.css.

You can use any of the components as demonstrated in the documentation. Please refer to each component's demo page to see how they should be imported.

It builds open source and commercial tools used by many hundreds of thousands of developers in production.

These cards below are built using MATERIAL-UI.

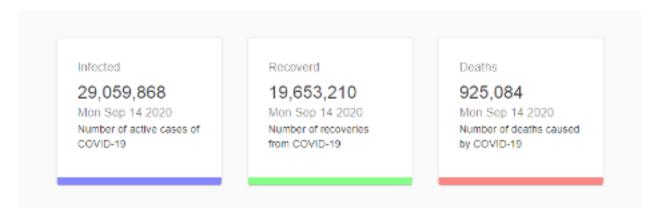
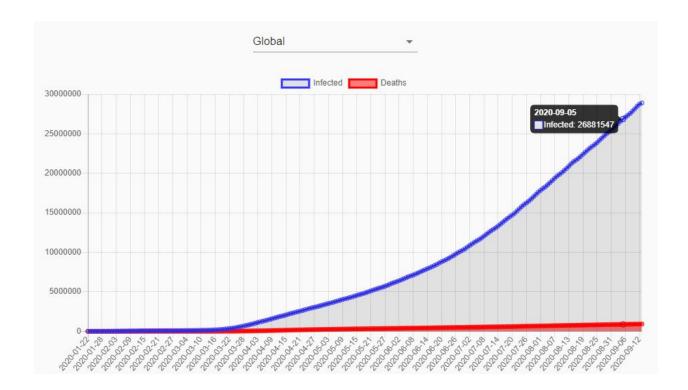
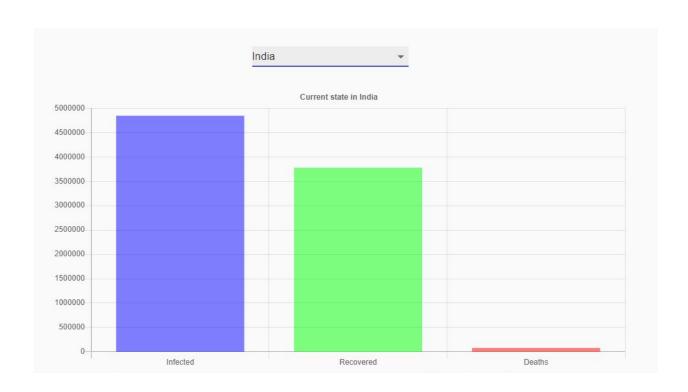


CHART-JS

Chart.js is a free open-source JavaScript library for data visualization, which supports 8 chart types: bar, line, area, pie (doughnut), bubble, radar, polar, and scatter. Created by London-based web developer Nick Downie in 2013, now it is maintained by the community and is the second most popular JS charting library on GitHub by the number of stars after D3.js, considered significantly easier to use though less customizable than the latter. Chart.js renders in HTML5 canvas and is widely covered as one of the best data visualization libraries. It is available under the MIT license

Usage of chart.js





• An application programming interface is a computing interface which defines interactions

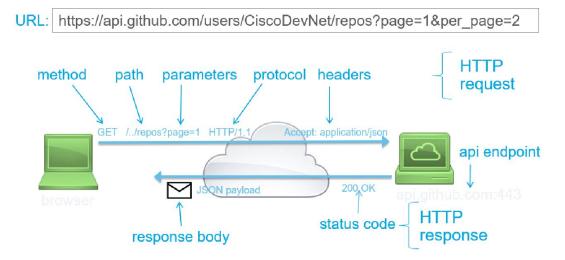
between multiple software intermediaries.

- It defines the kinds of calls or requests that can be made, how to make them, the data formats that should be used, the conventions to follow.
- The API used in this project:

https://covid19.mathdro.id/api

As best place and practice for external API calls is React Lifecycle method componentDidMount(), where after the execution of the API call you should update the local state to be triggered new render() method call, then the changes in the updated local state will be applied on the component view.

Anatomy of a REST API query



IMPLEMENTATION

```
EXPLORER
                          index.js
> OPEN EDITORS
                          src > api > 📻 index.js > ...
                                 import axios from "axios";
CORONA TRACKER
 node_modules
                                 const url = "https://covid19.mathdro.id/api";

∨ □ public

                                 export const fetchData = async (country) => [
     favicon.ico
                                   let changeableUrl = url;
     index.html
     logo192.png
                                   if (country) {
                                     changeableUrl = `${url}/countries/${country}`;
     logo512.png
     ⟨→ manifest.json
                                   try {
     robots.txt
                                     const (
                                       data: { confirmed, recovered, deaths, lastUpdate },
 V FRE STO
                                     } = await axios.get(changeableUrl);
  V 📺 api
                                     return ( confirmed, recovered, deaths, lastUpdate );
   index.is
                                   } catch (error) {
                                     console.log(error);
  components
   V Cards
       Cards.jsx
                                 export const fetchDailyData = async () => (
       Cards.module...
                                   try {
   ∨ E Chart
                                     const ( data ) = await axios.get(`${url}/daily`);
                                     const modifiedData = data.map((dailyData) => ({
       Chart.jsx
                                       confirmed: dailyData.confirmed.total,
       Chart.module...
                                       deaths: dailyData.deaths.total,

✓ Im CountryPicker

                                       date: dailyData.reportDate,
                                     }));
       CountryPicke...
                                     return modifiedData;
       CountryPicke...
                                   } catch (error) {
      index.js
                                     console.log(error);
  > images
     App.js
                                 export const fetchCountries = async () => [
     App.module.css
                                   try {
     index.js
                                     const {

    gitignore

                                      data: { countries },
                                     } = await axios.get(`${url}/countries`);
    package-lock.json
                                     return countries.map((country) => country.name);
    package.json
                                    } catch (error) {
    README.md
                                     console.log(error);
OUTLINE
                           44
> TIMELINE
```

```
EXPLORER
                          > OPEN EDITORS
                          src > components > Cards > @ Cards.jsx > [@] Cards
                                 import React from "react";
V CORONA TRACKER
                                 import styles from "./Cards.module.css";
  > node_modules
                                 import ( Card, CardContent, Grid, Typography ) from "@material-ui/core";
  V 📠 public
                                 import CountUp from "react-countup";
                                 import cx from "classnames";
      * favicon.ico
      index.html
                                 const Cards = ({ data: { confirmed, recovered, deaths, lastUpdate } }) => {
                                   if (!confirmed) {
      logo192.png
                                     return "Loading ... ";
      logo512.png
     ( manifest.json
                                   return (
                                     <div className={styles.container}>
      robots.txt
                                       <Grid container spacing={3} justify="center">
  V TO SIC
                                         <Grid
   V 🚎 api
                                           item
                                           component={Card}
       index.js
                                           xs = \{12\}

✓ I components

                                           md=\{3\}
    ∨ 庙 Cards
                                           className={cx(styles.card, styles.infected)}
        Cards.jsx
                                           <CardContent>
        Cards.module...
                                             <Typography color="textSecondary" gutterBottom>
                                              Infected
    ∨ 🗁 Chart
                                             </Typography>
        Chart.jsx
        <Typography variant="h5">
                                               <CountUp
    ∨ E CountryPicker
                                                start={0}
        CountryPicke...
                                                end={confirmed.value}
        3 CountryPicke...
                                                duration=(2.5)
                                                separator=","
       index.js
                                               1>
    > images
                                             </Typography>
      App.js
                                             <Typography color="textSecondary">
      App.module.css
                                              (new Date(lastUpdate).toDateString())
      index.js
                                             </Typography>
     o .gitignore
                                             <Typography variant="body2">
     package-lock.json
                                             Number of active cases of COVID-19
                                             </Typography>
     @ package.json
                                           </CardContent>
     A DEADME and
                                         </Grid>
> OUTLINE
> TIMELINE
                                         <Grid
```

```
EXPLORER
                          Cards.jsx X
                           src > components > Cards > 🚳 Cards.jsx > 📵 Cards
> OPEN EDITORS
V CORONA TRACKER
                                          <Grid
                                            item
  > node modules
                                            component={Card}
 V 📠 public
                                            xs = \{12\}
     * favicon.ico
                                            md=\{3\}
                                            className={cx(styles.card, styles.recovered)}
     index.html
     logo192.png
                                            <CardContent>
                                              <Typography color="textSecondary" gutterBottom>
     logo512.png
                                              Recoverd
     ( manifest.json
                                              </Typography>
     robots.txt
                                              <Typography variant="h5">
 V FR SIC
                                                <CountUp
  V 😇 api
                                                  start={0}
      index.js
                                                  end={recovered.value}
                                                  duration={2.5}

✓ I components

                                                  separator=","
    ∨ 庙 Cards
                                                1>
                                              </Typography>
        Cards.jsx
        Cards.module...
                                              <Typography color="textSecondary">
    ∨ 🗁 Chart
                                                {new Date(lastUpdate).toDateString()}
                                              </Typography>
        Chart.jsx
        Chart.module...
                                              <Typography variant="body2">
    ∨ CountryPicker
                                              Number of recoveries from COVID-19
                                              </Typography>

    CountryPicke...

                                            </CardContent>
        CountryPicke...
                                          </Grid>
       index.js
                                          <Grid
   > images
                                            item
     App.js
                                            component={Card}
                                            xs = \{12\}
     App.module.css
                                            md = \{3\}
     index.js
                                            className={cx(styles.card, styles.deaths)}
    o .gitignore
                                            <CardContent>
    package-lock.json
                                              <Typography color="textSecondary" gutterBottom>
    package.json
                                               Deaths
    A DEADME and
                                              </Typography>
OUTLINE
                                              <Typography variant="h5">
> TIMELINE
```

```
Cards.jsx
  EXPLORER
> OPEN EDITORS
                          src > components > Cards > @ Cards.jsx > [@] Cards
V CORONA_TRACKER
                                             <Typography variant="h5">
  > node_modules
                                               <CountUp
                                                 start={0}
 v 📻 public
                                                 end={deaths.value}
     favicon.ico
                                                 duration=\{2.5\}
                                                 separator=","
     index.html
                                               1>
     logo192.png
                                             </Typography>
     logo512.png
                                             <Typography color="textSecondary">
     manifest.json
                                               {new Date(lastUpdate).toDateString()}
     robots.txt
                                             </Typography>
 V STC
                                             <Typography variant="body2">
  v 🚎 api
                                               Number of deaths caused by COVID-19
      index.js
                                             </Typography>
                                           </CardContent>

✓ III components

                                         </Grid>
    ∨ To Cards
                                       </Grid>
       Cards.jsx
                                     </div>
                                   );
       Cards.module...
                                 ];
    ∨ 🗁 Chart
       Chart.jsx
                                 export default Cards;
       Chart.module...
    ∨ 📻 CountryPicker
       CountryPicke...
       3 CountryPicke...
      index.js
   > images
     App.js
     App.module.css
     index.js
    gitignore
    package-lock.json
    package.json
    A DEADLAE and
OUTLINE
> TIMELINE
```

```
EXPLORER
                         Cards.module.css ×
> OPEN EDITORS
                          src > components > Cards > 3 Cards.module.css > 4 .container
                                .container {
V CORONA TRACKER
                                  margin: 50px 0px;
  > node_modules
 V 📠 public
                                 .card {
                                  margin: 0 2% !important;
     * favicon.ico
     index.html
                                 .infected {
                                  border-bottom: 10px solid □rgba(0, 0, 255, 0.5);
     logo192.png
     logo512.png
                                .recovered {
     ( manifest.json
                                  border-bottom: 10px solid □rgba(0, 255, 0, 0.5);
     robots.txt
                                 .deaths {
 V FOR STO
                                  border-bottom: 10px solid ■rgba(255, 0, 0, 0.5);
  v 😇 api
      index.js
                                @media (max-width: 770px) {
  components
                                   .card (
    ∨ 🔚 Cards
                                   margin: 2% 0 !important;
       Cards.jsx
       3 Cards.module...
    ∨ 🗁 Chart
       Chart.jsx

✓ Image: CountryPicker

       CountryPicke...
       3 CountryPicke...
      index.js
   > images
     App.js
     App.module.css
     index.js

    gitignore

    package-lock.json
    package.json
    A DEADME and
> OUTLINE
> TIMELINE
```

```
EXPLORER
                         > OPEN EDITORS
                          src > components > Chart > @ Chart.jsx > ...
                                 import React, { useEffect, useState } from "react";
V CORONA TRACKER
                                 import ( fetchDailyData ) from "../../api";
  > node_modules
                                 import ( Line, Bar ) from "react-chartjs-2";
 v le public
                                 import styles from "./Chart.module.css";
     * favicon.ico
                                 const Chart = ({ data: { confirmed, recovered, deaths }, country }) => {
     index.html
                                   const [dailyData, setDailyData] = useState([]);
     logo192.png
                                   useEffect(() => {
     logo512.png
                                     const fetchAPI = async () => {
     ( manifest.json
                                      setDailyData(await fetchDailyData());
     robots.txt
 V FRE STO
                                    fetchAPI();
  V 📴 api
                                   }, []);
      index.js
                                   const lineChart = dailyData.length ? (

✓ III components

                                     <Line
    ∨ E Cards
                                       data={{
                                         labels: dailyData.map(( date )) => date),
        Cards.jsx
                                         datasets:
        3 Cards.module...
                                            data: dailyData.map(({ confirmed }) => confirmed),
    V 🗁 Chart
                                            label: "Infected",
       Chart.jsx
                                            borderColor: "#3333ff",
        fill: true,
    ∨ 庙 CountryPicker

♠ CountryPicke...

                                            data: dailyData.map(({ deaths }) => deaths),
                                            label: "Deaths",
        CountryPicke...
                                            borderColor: "red",
      index.js
                                            backgroundColor: "rgba(255,0,0,0.5)",
   > images
                                            fill: true,
     App.js
     App.module.css
     index.js
                                     1>
                                   ) : null;
    .gitignore
    package-lock.json
                                   const barChart = confirmed ? (
                                     <Bar
    package.json
    A DEADME and
                                         labels: ["Infected", "Recovered", "Deaths"],
> OUTLINE
                                         datasets: [
> TIMELINE
```

```
EXPLORER
> OPEN EDITORS
                          src > components > Chart > @ Chart.jsx > ...
V CORONA TRACKER
                                  const barChart = confirmed ? (
  > node_modules
                                    <Bar
                                      data={{
 V P public
                                        labels: ["Infected", "Recovered", "Deaths"],
     * favicon.ico
                                        datasets: [
     index.html
                                            label: "People",
     logo192.png
                                            backgroundColor: |
     logo512.png
                                              "rgba(0, 0, 255, 0.5)",
     ( manifest.json
                                              "rgba(0, 255, 0, 0.5)",
                                             "rgba(255, 0, 0, 0.5",
     robots.txt
 V FOR STO
                                           data: [confirmed.value, recovered.value, deaths.value],
  v 👜 api
                                        ],
      index.js

∨ iii components

                                      options={{
                                        legend: { display: false },
   ∨ To Cards
                                        title: { display: true, text: 'Current state in ${country}' },

☆ Cards.jsx

       Cards.module...
                                    15
                                  ) : null;
    ∨ E Chart
       Chart.isx
                                  return (
                                  <div className=(styles.container)>(country ? barChart : lineChart)</div>
       );
    ∨ 🚡 CountryPicker
                                1;

₩ CountryPicke...

                                export default Chart;
       CountryPicke...
      index.js
   > images
     App.js
     App.module.css
     index.js
    .gitignore
    package-lock.json
    package.json
    A DEADME and
OUTLINE
> TIMELINE
```

```
EXPLORER
                        Chart.module.css ×
                         src > components > Chart > 3 Chart.module.css > ...
> OPEN EDITORS
                               .container {
V CORONA TRACKER
                                 display: flex;
 > node modules
                                 justify-content: center;
 v 📠 public
                                 width: 85%;
     * favicon.ico
                               @media (max-width: 770px) {
     index.html
                                 .container {
     logo192.png
                                  width: 100%;
     logo512.png
    ( ) manifest.json
                          11
     robots.txt
 V FR SIC
  V 🕮 api
     index.js
  components
   ∨ Cards
       Cards.jsx
       Cards.module...
    V 🗁 Chart
       Chart.jsx
       ∨ CountryPicker
       CountryPicke...
       CountryPicke...
      index.js
   > images
     App.js
     App.module.css
     index.js
   .gitignore
    package-lock.json
    package.json
```

A DEADME

> OUTLINE > TIMELINE

```
EXPLORER
                          CountryPicker.jsx ×
> OPEN EDITORS
                           src > components > CountryPicker > 🕸 CountryPicker.jsx > ...
                                 import React, { useEffect, useState } from "react";
V CORONA TRACKER
                                  import ( NativeSelect, FormControl ) from "@material-ui/core";
  > node_modules
 v 📠 public
                                  import styles from "./CountryPicker.module.css";
                                  import { fetchCountries } from "../../api";
     favicon.ico
      index.html
                                  const CountryPicker = ({ handleCountryChange }) => {
                                    const [fetchedCountries, setFetchedCountries] = useState([]);
     logo192.png
     logo512.png
                                   useEffect(() => {
     ( manifest.json
                                     const fetchAPI = async () => (
                                       setFetchedCountries(await fetchCountries());
     robots.txt
                                      1;
 V FRE STO
                                     fetchAPI();
   V 🔯 api
                                    }, [setFetchedCountries]);
      index.js
                                    return (

∨ I components

                                     <div>
    ∨ 庙 Cards
                                       <FormControl className={styles.formControl}>
                                          <NativeSelect
       Cards.jsx
                                           defaultValue=""
        Cards.module...
                                           onChange={(e) => handleCountryChange(e.target.value)}
    ∨ 🗁 Chart
                                           <option value="">Global</option>
       Chart.jsx
                                           {fetchedCountries.map((country, i) => (
        <option key={i} value={country}>
                                               {country}

✓ Image: CountryPicker

                                             </option>
       CountryPicke...
                                           ))]
        3 CountryPicke...
                                         </NativeSelect>
                                       </FormControl>
      index.js
                                     </div>
   > images
                                    );
                                 };
     App.js
     App.module.css
                                 export default CountryPicker;
     index.js
    o .gitignore
    package-lock.json
    package.json
    A DEADME and
> OUTLINE
> TIMELINE
```

```
EXPLORER
                         CountryPicker.jsx
                                               CountryPicker.module.css X
                         src > components > CountryPicker > 3 CountryPicker.module.css > 4 .formControl
> OPEN EDITORS
                                .formControl {
V CORONA TRACKER
                                 width: 90%;
 > node_modules
                                 margin-bottom: 30px !important;
 v 📠 public
     * favicon.ico
     index.html
     logo192.png
     logo512.png
     ⟨→⟩ manifest.json
     robots.txt
 V SIC
  V 😇 api
      index.js

✓ iii components

   ∨ Cards

♠ Cards.jsx

       Cards.module...
   ∨ 🗁 Chart
       Chart.jsx
       ∨ 🗁 CountryPicker

    CountryPicke...

       3 CountryPicke...
      index.js
   > images
     App.js
     3 App.module.css
     index.js
    gitignore
    package-lock.json
    package.json
    A DEADNAE and
OUTLINE
> TIMELINE
```



```
App.js
  EXPLORER
                                       X
                          src > 5 App.js > ...
> OPEN EDITORS
                                 import React, { Component } from "react";
V CORONA TRACKER
  > node modules
                                 import ( Cards, Chart, CountryPicker ) from "./components";
 ∨ m public
                                 import styles from "./App.module.css";
     * favicon.ico
                                 import { fetchData } from "./api";
     index.html
                                 import coronalmage from "./images/image.png";
     logo192.png
     logo512.png
                                 export class App extends Component (
                                   state = {
     manifest,json
                                     data: {},
     robots.txt
                                     country: "",
                                   1;
 V FOR STC
                                   async componentDidMount() {
   v 📴 api
                                     const fetchedData = await fetchData();
      index.js
                                     this.setState([ data: fetchedData ]);
   Components

✓ ☐ Cards

                                   handleCountryChange = async (country) => {
       Cards.jsx
                                     const fetchedData = await fetchData(country);
                                     this.setState({ data: fetchedData, country: country });
        Cards.module...
                                   };
    ∨ 🗁 Chart
                                   render() {
       Chart.jsx
                                     const { data, country } = this.state;

∃ Chart.module...

                                     return (

✓ Im CountryPicker

                                       <div className={styles.container}>
                                         <img className={styles.image} alt="corona" src={coronaImage} />
       CountryPicke...
                                         <Cards data={data} />
        CountryPicke...
                                         <CountryPicker handleCountryChange={this.handleCountryChange} />
      index.js
                                         <Chart data={data} country={country} />
                                       </div>
   > images
                                     );
     App.js
     App.module.css
     index.js
                                 export default App;
    gitignore
    package-lock.json
    package.json
    README.md
> OUTLINE
> TIMELINE
```

```
App.module.css ×
  EXPLORER
                          src > 3 App.module.css > 2 body
> OPEN EDITORS
                                 body {
V CORONA TRACKER
                                   background-color: ☐rgb(250, 250, 250);
  > node_modules
 ∨ E public
                                 .container {
     * favicon.ico
                                   display: flex;
                                   justify-content: center;
     index.html
                                   align-items: center;
     logo192.png
                                   flex-direction: column;
     logo512.png
                                 .image {
     {_} manifest.json
                                   width: 370px;
     robots.txt
                                   margin-top: 50px;
    FEE STO
                                 @media (max-width: 770px) (
  v 📴 api
                                   .container {
      index.js
                                   margin: 0 10%;
  components
                                   .image {

✓ ☐ Cards

                                   width: 100%;
       Cards.jsx
       Cards.module...
    ∨ 庙 Chart
       Chart.jsx
       3 Chart.module...

✓ Image: CountryPicker

       CountryPicke...
       CountryPicke...
     index.js
   > images
     App.js

₱ App.module.css

     index.js
    gitignore
    package-lock.json
    package.json
    README.md
OUTLINE
> TIMELINE
```

```
3 App.module.css
  EXPLORER
                                               index.js
                                                            X
> OPEN EDITORS
                          src > 5 index.js
                                 import React from "react";
V CORONA TRACKER
                                 import ReactDOM from "react-dom";
  > node_modules
                                 import App from "./App";
 ∨ m public
     * favicon.ico
                                 ReactDOM.render(<App />, document.getElementById("root"));
     index.html
     logo192.png
     logo512.png
     manifest,json
     robots.txt
    FOE STO
   v 📴 api
      index.js

✓ I components

✓ ☐ Cards

       Cards.jsx
       3 Cards.module...
    ∨ 庙 Chart
       ♦ Chart.jsx
       3 Chart.module...

✓ Image: CountryPicker

♠ CountryPicke...

       CountryPicke...
      index.js
   > images
     App.js
     App.module.css
    index.js
    gitignore
    package-lock.json
    package.json
    README.md
> OUTLINE
> TIMELINE
```

```
EXPLORER
                          package.json X
> OPEN EDITORS
                           package.json > ...
V CORONA TRACKER
                                    "name": "corona_tracker",
  > node modules
                                    "version": "0.1.0",
 V public
                                    "private": true,
                                    "dependencies": {
     * favicon.ico
                                      "@material-ui/core": "^4.11.0",
     index.html
                                      "@testing-library/jest-dom": "^4.2.4",
                                      "@testing-library/react": "^9.5.0",
     logo192.png
                                      "@testing-library/user-event": "^7.2.1",
     logo512.png
                                      "axios": "^0.20.0",
     ( ) manifest ison
                                      "chart.js": "^2.9.3",
                                      "classnames": "^2.2.6",
     robots.txt
                                      "react": "^16.13.1",
 V FR SIC
                                      "react-chartjs-2": "^2.10.0",
  v 📴 api
                                      "react-countup": "^4.3.3",
                                      "react-dom": "^16.13.1",
      index.js
                                      "react-scripts": "3.4.3"
  components
                                    ▶ Debug
    ∨ Fards
                                    "scripts": {
       Cards.jsx
                                      "start": "react-scripts start",
        Cards.module...
                                      "build": "react-scripts build",
                                      "test": "react-scripts test",
    ∨ E Chart
                                      "eject": "react-scripts eject"
       ♦ Chart.jsx
        Chart.module...
                                    "eslintConfig": {
                                      "extends": "react-app"

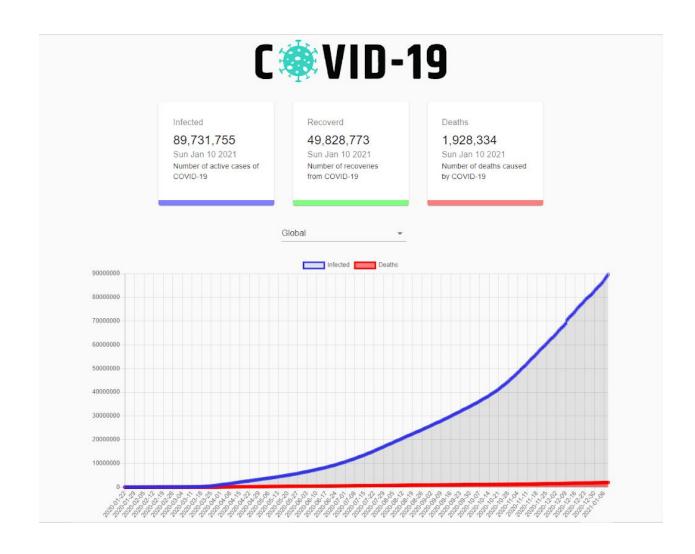
✓ Im CountryPicker

       CountryPicke...
                                    "browserslist": {
        CountryPicke...
                                      "production": [
                                        ">0.2%".
      index.js
                                       "not dead",
   > images
                                        "not op mini all"
     App.js
                                      "development":
     App.module.css
                                       "last 1 chrome version",
     index.js
                                       "last 1 firefox version",
                                        "last 1 safari version"
    .gitignore
    package-lock.json
    package.json
                            41
    A DEADMAR and
OUTLINE
> TIMELINE
```

RESULT ANALYSIS

We have come to some results which state the functioning of Covid-19 detector in its full potential. So the properties of the result we get after analysing are:

- Quick communication with visitors.
- Digital growth.
- Providing a fully developed website and app.
- General awareness
- Graphical content



CONCLUSION, LIMITATION AND FUTURE SCOPE

CONCLUSION

In this overall conclusion of the project the website will provide according to the given command and can easily work as the instruction which is given by the customer. The website is designed in such a way that it can provide customised website, advertisement, content writing, according to the requirement and information provided.

LIMITATION

After the completion of the overall project still the project contains some of the limitations but such limitations can be solved if our project is raised to the higher levels. The limitation are as follows:

- The involvement of technology in the current project is very low.
- There is no option to classify between the genuine and fake or fraud customer, which increases the probability of fraud and also increases another risk.
- The user interface of the website is also a limitation.

FUTURE SCOPE

The future scope of our project will be very useful in various fields when the system is enhanced as per the requirements for the relevant field. Here we have the main concern to enhance the system:

• Machine learning and other advanced techniques can be applied to improve the quality of content writing or even create the content there on.

- The involvement of data analytics can provide a more customised result to our client.
- A better animation and good user interface used graphics can be added to enhance the user intake of the website.

REFERENCES

[1]Learning React: Functional Web Development with React and Redux by Alex Banks & Eve Porcello

[2] The Road to React by Robin Wieruch

[3]https://www.w3schools.com/react

[4]https://www.udemy.com/course/modern-react-front-to-back

[5]https://covid19.who.int