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

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

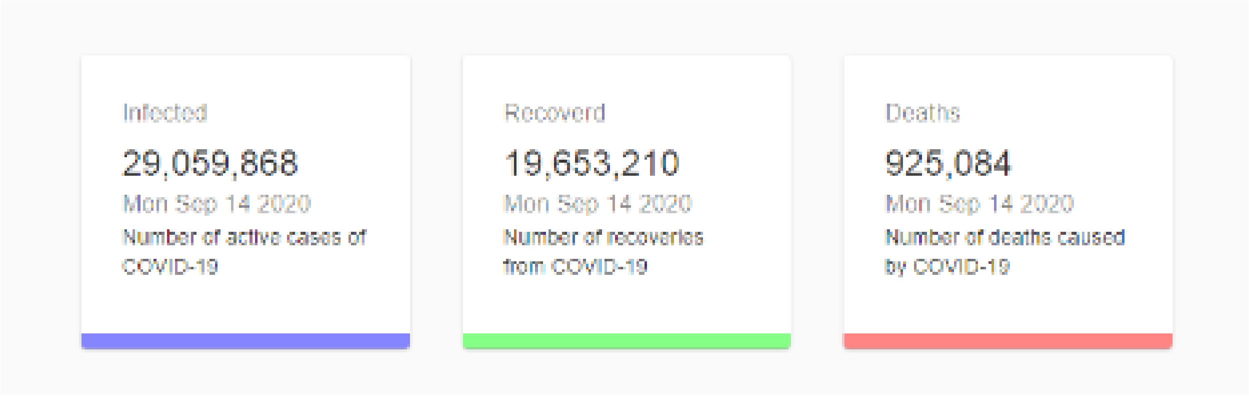
Material-UI provides an optional CssBaseline component. It fixes some inconsistencies across browsers and devices while providing slightly more opinionated resets to common HTML elements.

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.



Infected

29,059,868

Mon Sep 14 2020

Number of active cases of COVID-19

Recoverd

19,653,210

Mon Sep 14 2020

Number of recoveries from COVID-19

Deaths

925,084

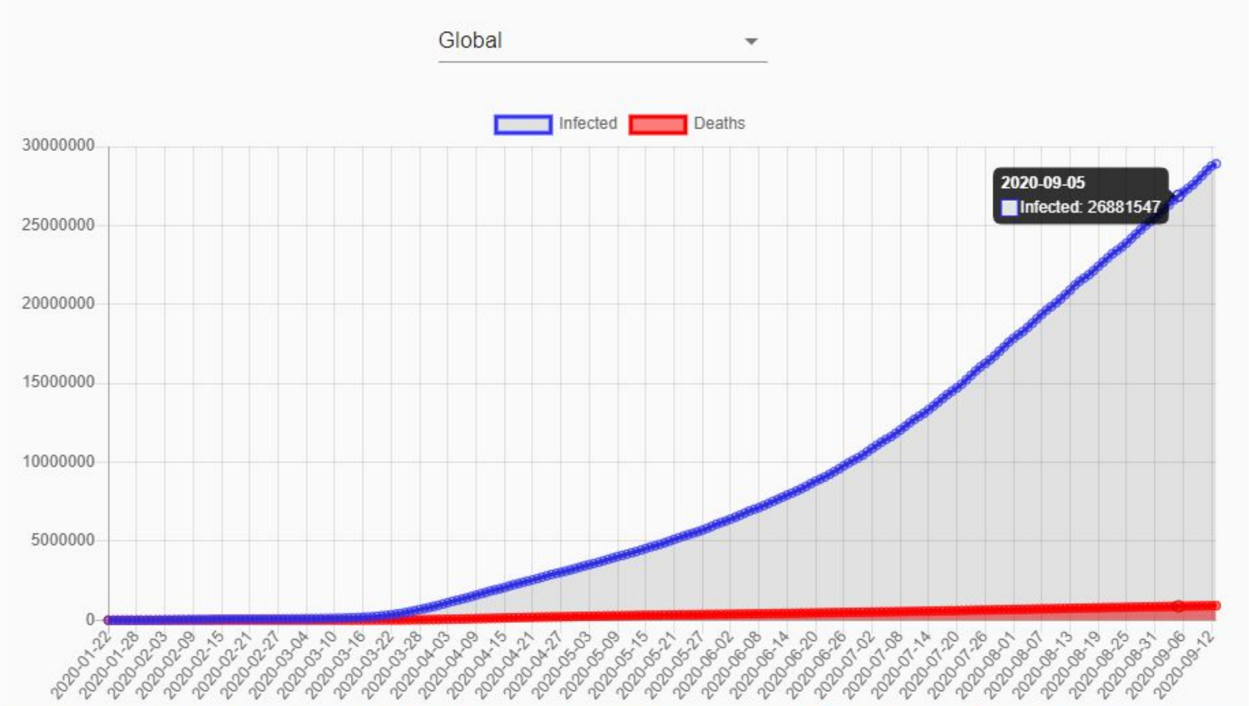
Mon Sep 14 2020

Number of deaths caused by COVID-19

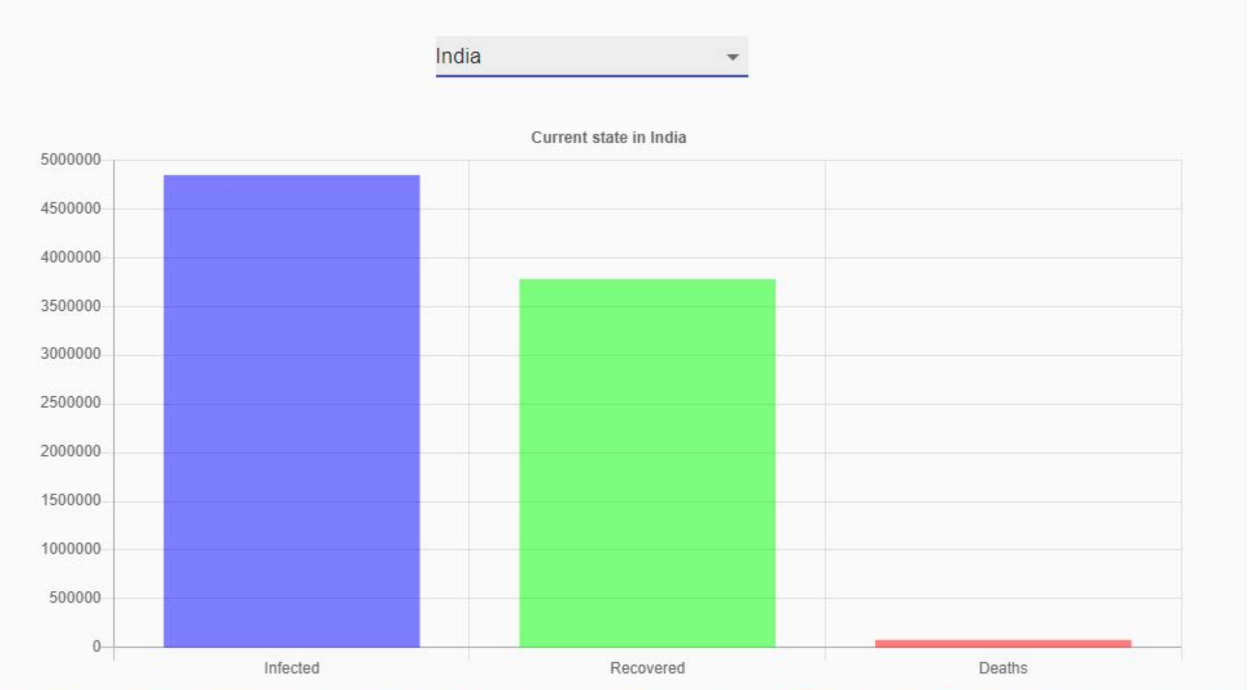
CHART-JS

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



Graph for country-wise data



India

Current state in India
5000000
4500000
4000000
3500000
3000000
2500000
2000000
1500000
1000000
500000
0
Infected
Recovered
Deaths

API

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

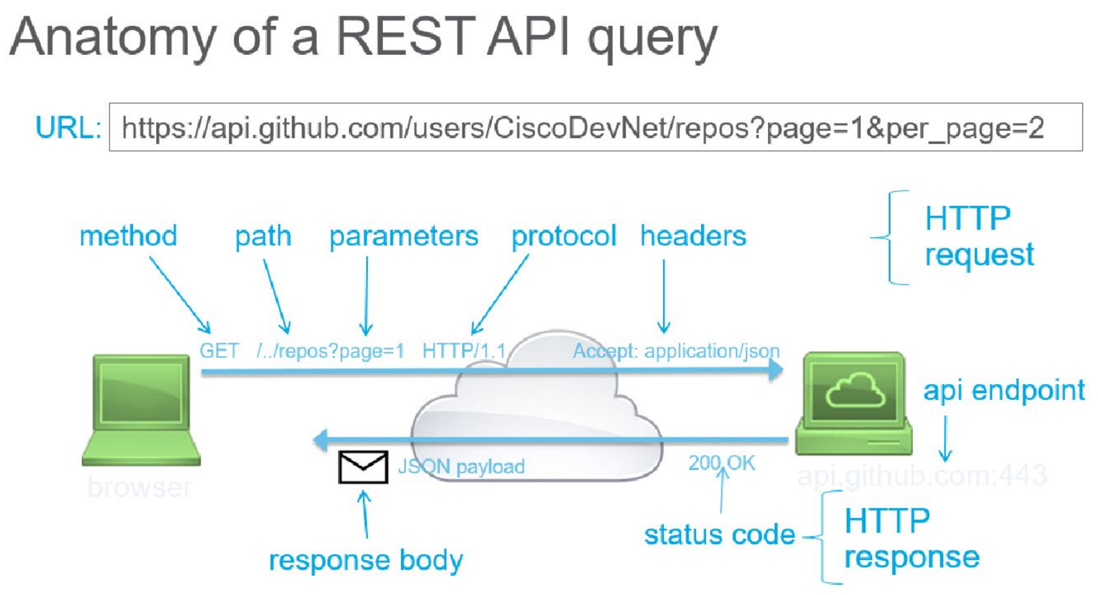
between multiple software intermediaries.

- 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.

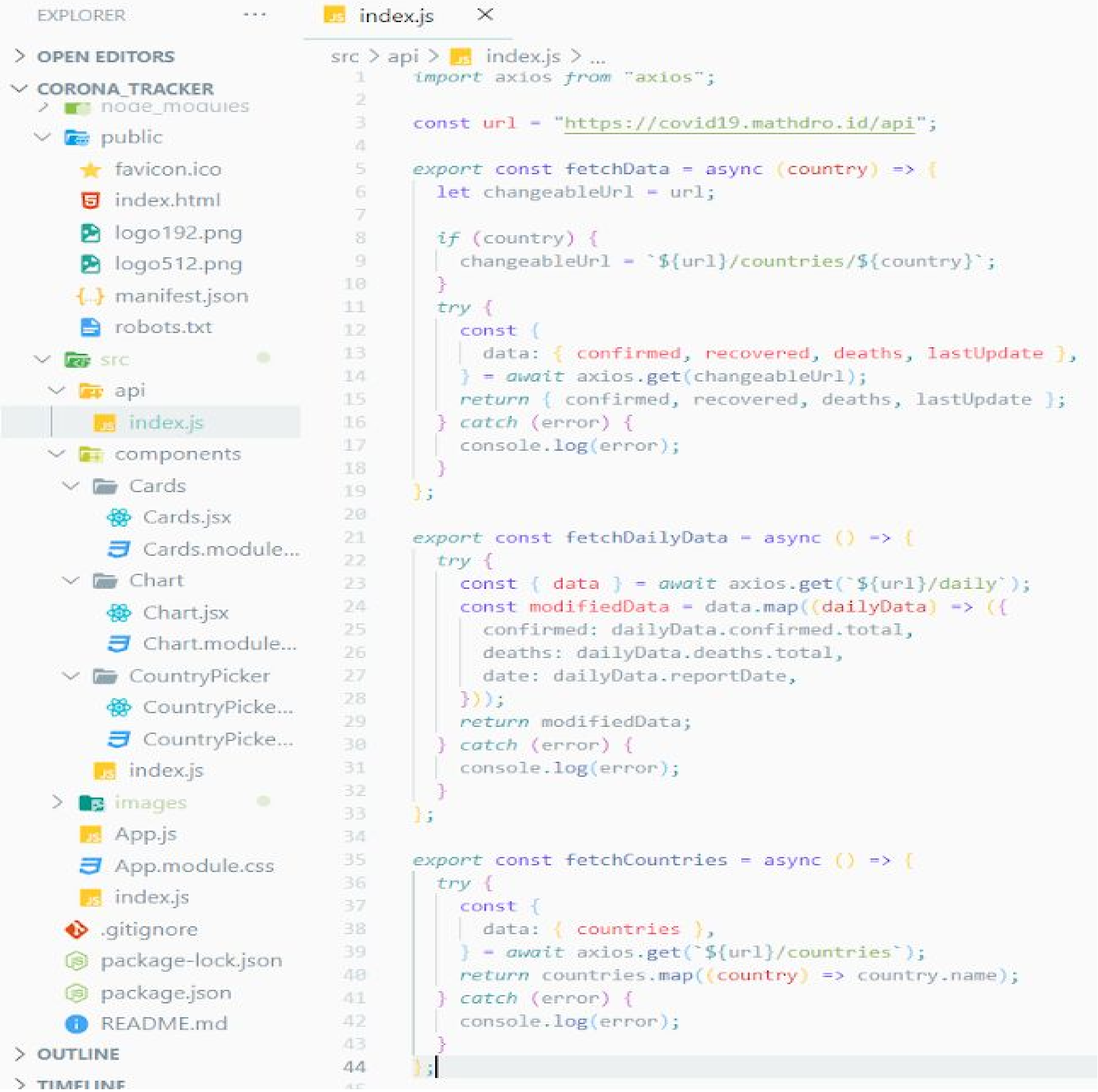
- The API used in this project:

https://covid19.mathdro.id/api

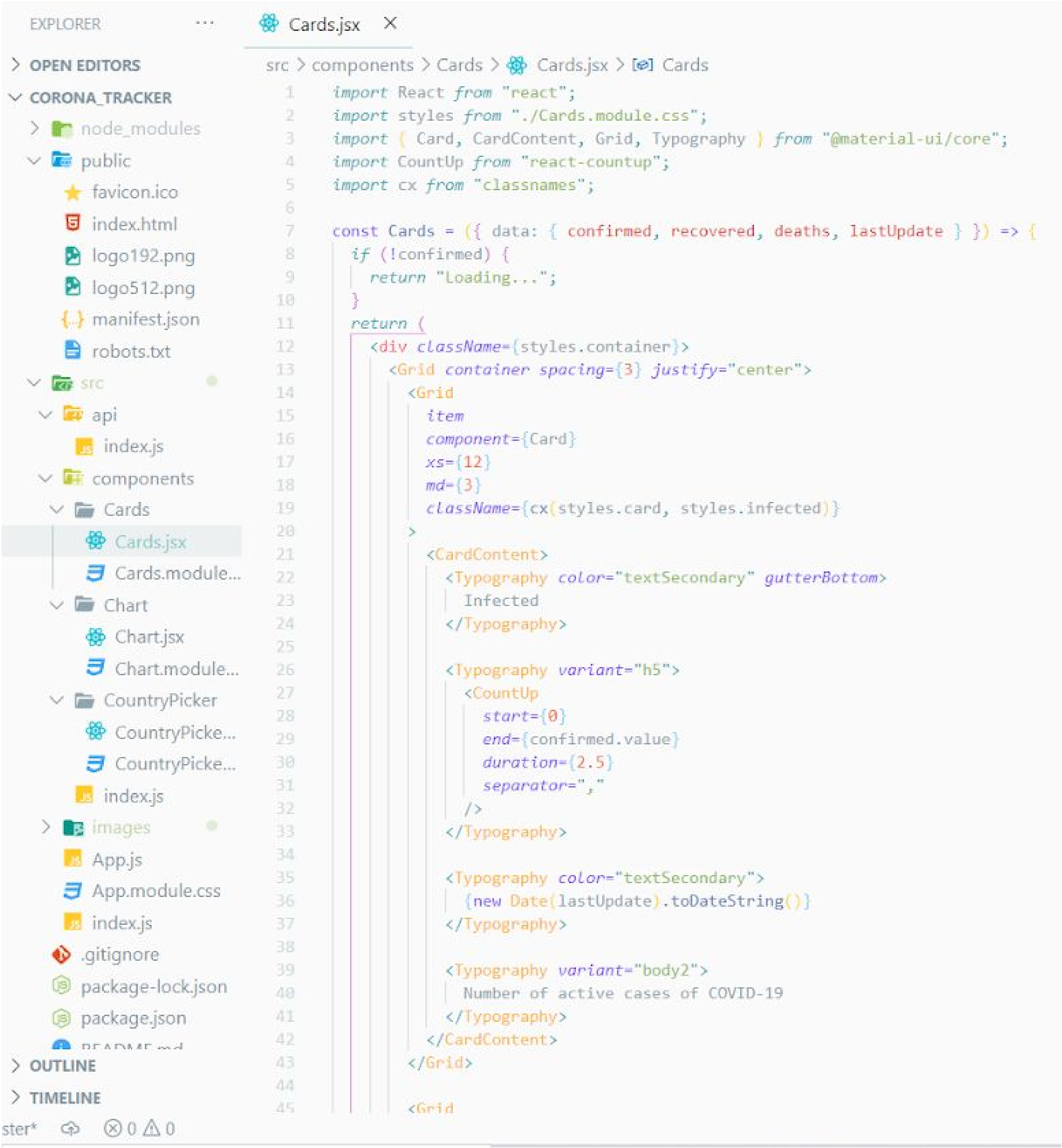
As best place and practice for external API calls is React Lifecycle method componentDidMount(), where after the execution of the API call you should update the local state to be triggered new render() method call, then the changes in the updated local state will be applied on the component view.



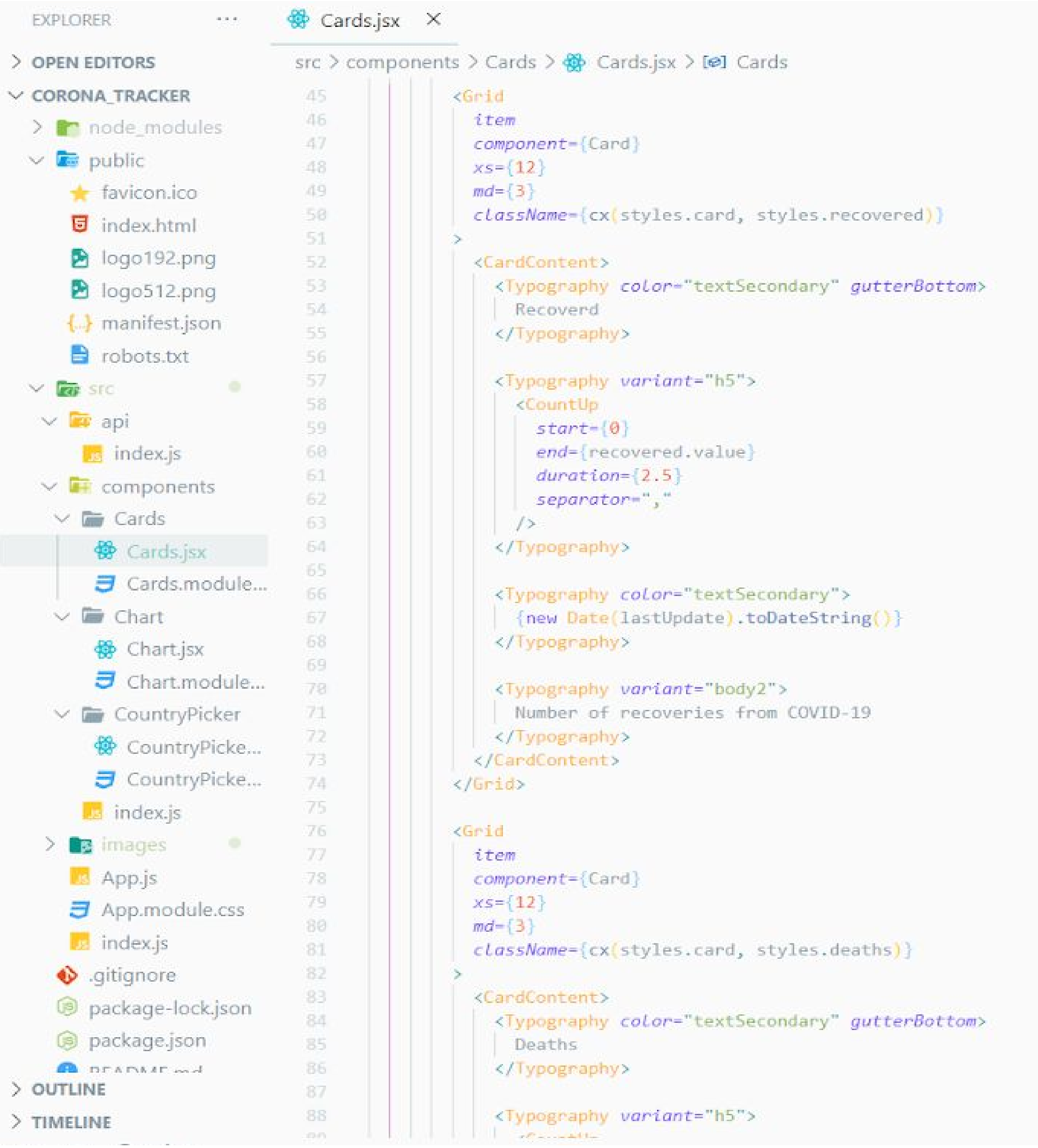
IMPLEMENTATION



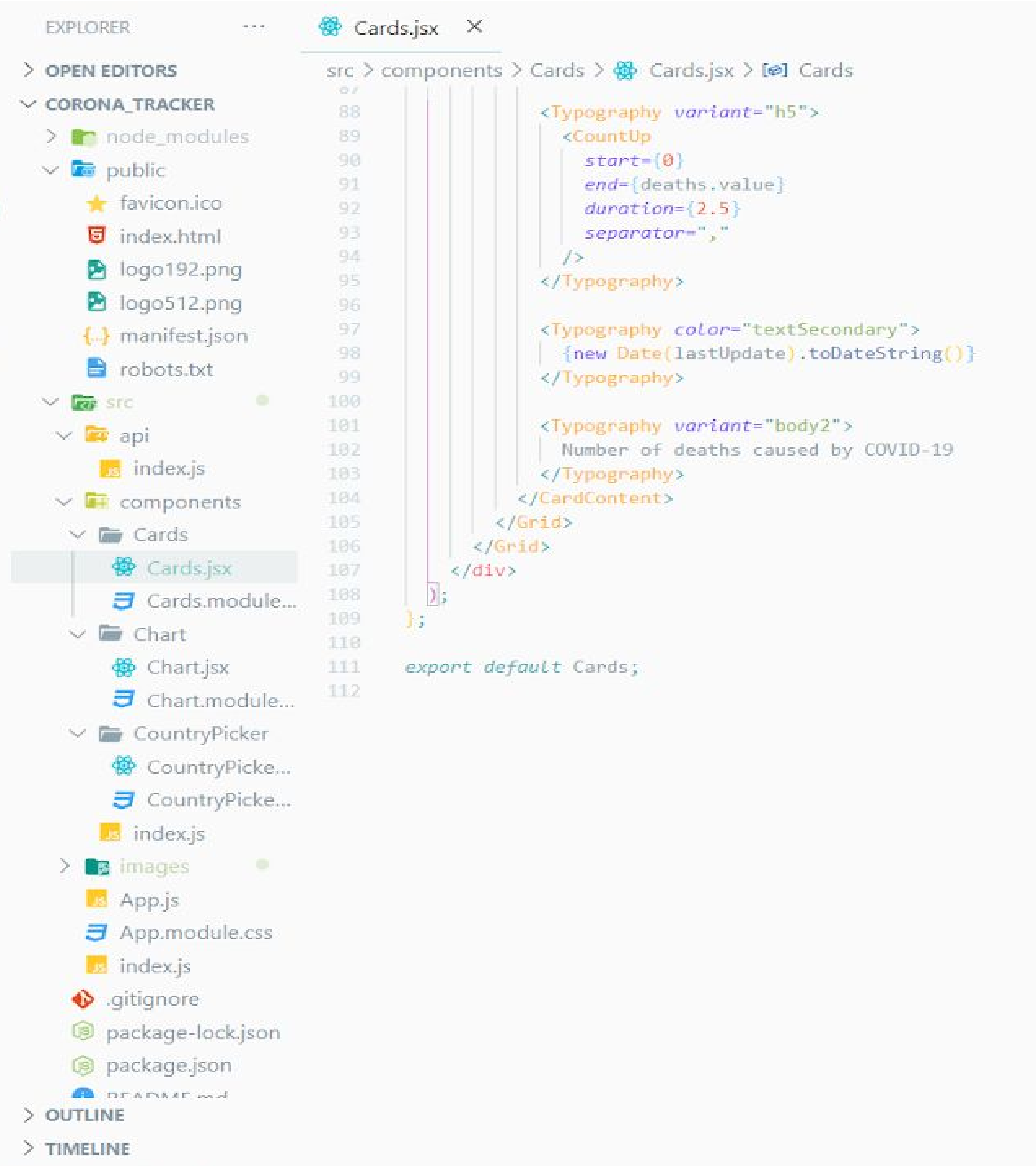
```
import axios from "axios";

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

export const fetchData = async (country) => {
  let changeableUrl = url;

  if (country) {
    changeableUrl = `${url}/countries/${country}`;
  }
  try {
    const {
      data: { confirmed, recovered, deaths, lastUpdate },
    } = await axios.get(changeableUrl);
    return { confirmed, recovered, deaths, lastUpdate };
  } catch (error) {
    console.log(error);
  }
};

export const fetchDailyData = async () => {
  try {
    const { data } = await axios.get(`${url}/daily`);
    const modifiedData = data.map((dailyData) => ({
      confirmed: dailyData.confirmed.total,
      deaths: dailyData.deaths.total,
      date: dailyData.reportDate,
    }));
    return modifiedData;
  } catch (error) {
    console.log(error);
  }
};

export const fetchCountries = async () => {
  try {
    const {
      data: { countries },
    } = await axios.get(`${url}/countries`);
    return countries.map((country) => country.name);
  } catch (error) {
    console.log(error);
  }
};
```



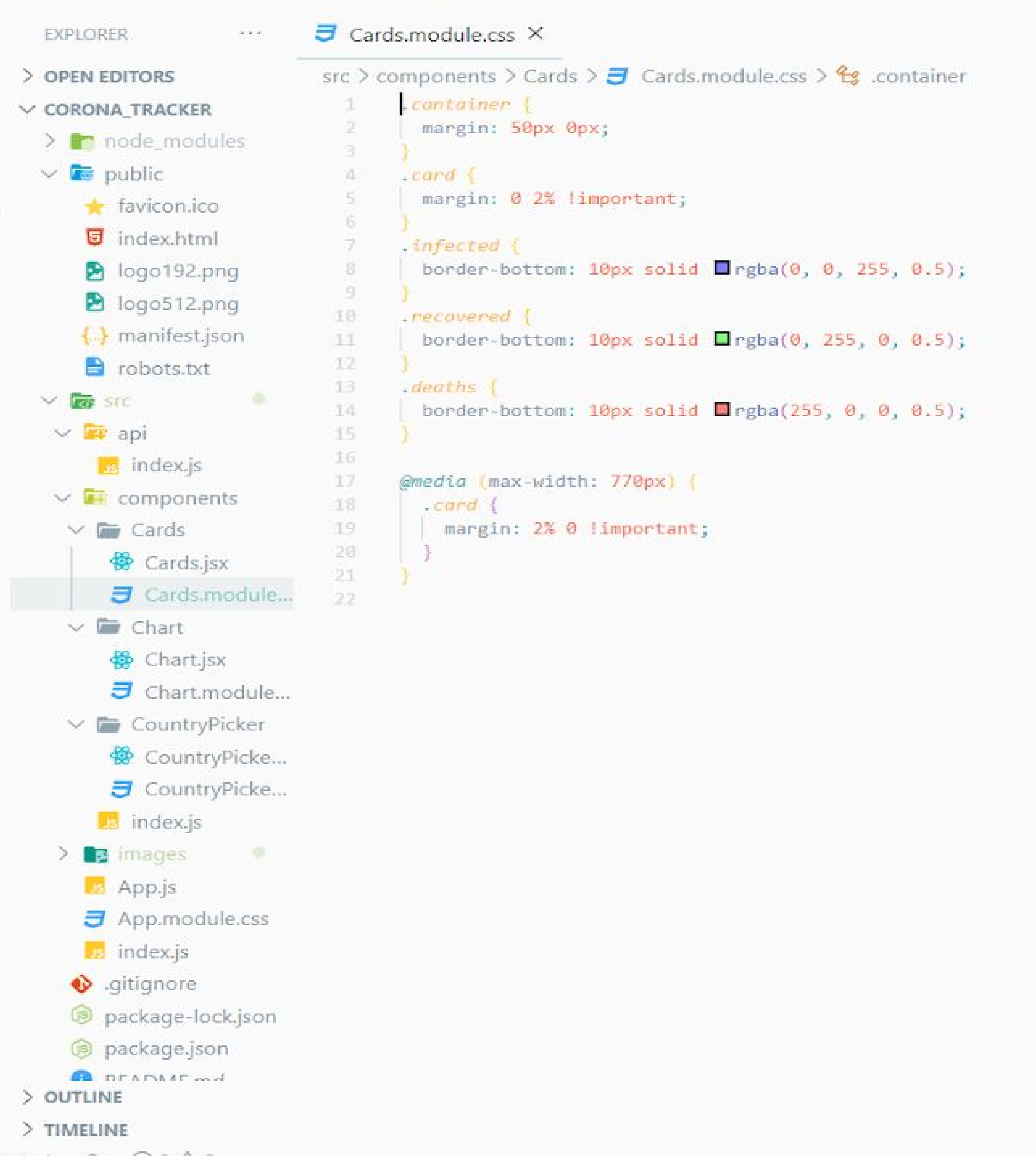
```jsx
import React from "react";
import styles from "./Cards.module.css";
import { Card, CardContent, Grid, Typography } from "@material-ui/core";
import CountUp from "react-countup";
import cx from "classnames";

const Cards = ({ data: { confirmed, recovered, deaths, lastUpdate } }) => {
  if (!confirmed) {
    return "Loading...";
  }
  return (
    <div className={styles.container}>
      <Grid container spacing={3} justify="center">
        <Grid
          item
          component={Card}
          xs={12}
          md={3}
          className={cx(styles.card, styles.infected)}
        >
          <CardContent>
            <Typography color="textSecondary" gutterBottom>
              Infected
            </Typography>

            <Typography variant="h5">
              <CountUp
                start={0}
                end={confirmed.value}
                duration={2.5}
                separator=","
              />
            </Typography>

            <Typography color="textSecondary">
              {new Date(lastUpdate).toDateString()}
            </Typography>

            <Typography variant="body2">
              Number of active cases of COVID-19
            </Typography>
          </CardContent>
        </Grid>

        <Grid
```



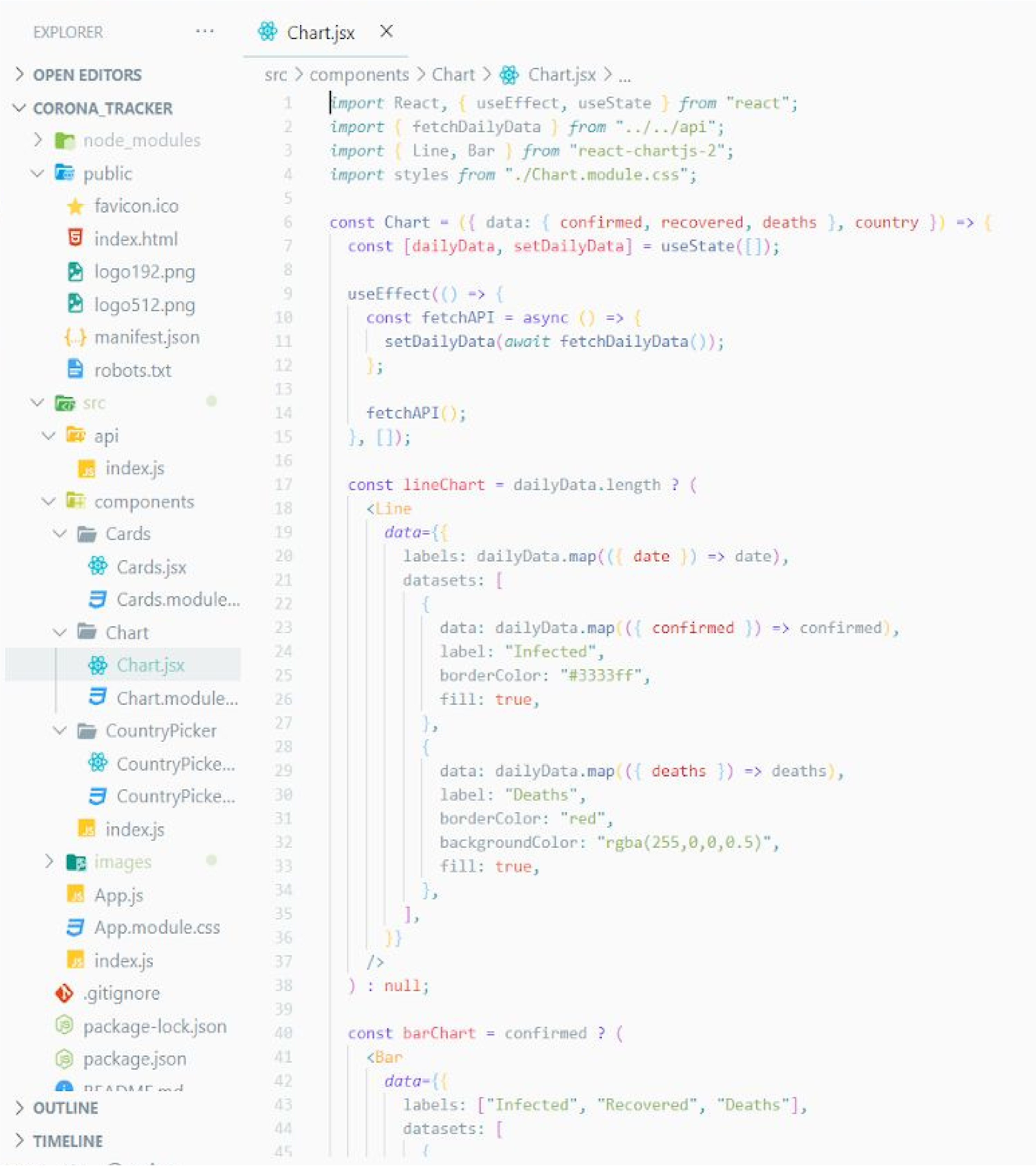
```
        <Grid
          item
          component={Card}
          xs={12}
          md={3}
          className={cx(styles.card, styles.recovered)}
        >
          <CardContent>
            <Typography color="textSecondary" gutterBottom>
              Recoverd
            </Typography>

            <Typography variant="h5">
              <CountUp
                start={0}
                end={recovered.value}
                duration={2.5}
                separator=","
              />
            </Typography>

            <Typography color="textSecondary">
              {new Date(lastUpdate).toDateString()}
            </Typography>

            <Typography variant="body2">
              Number of recoveries from COVID-19
            </Typography>
          </CardContent>
        </Grid>

        <Grid
          item
          component={Card}
          xs={12}
          md={3}
          className={cx(styles.card, styles.deaths)}
        >
          <CardContent>
            <Typography color="textSecondary" gutterBottom>
              Deaths
            </Typography>

            <Typography variant="h5">
              <CountUp
```



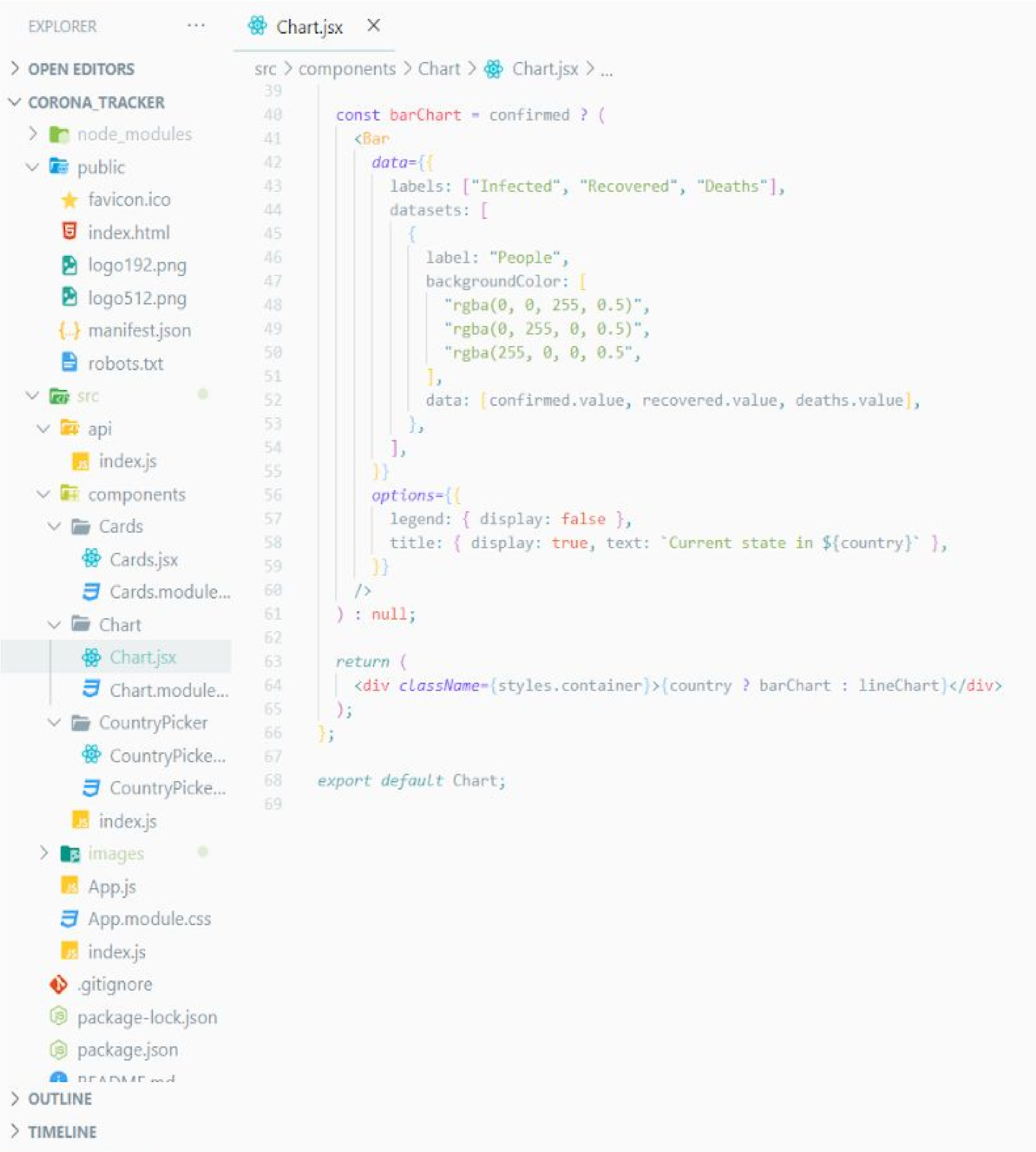
```
            <Typography variant="h5">
              <CountUp
                start={0}
                end={deaths.value}
                duration={2.5}
                separator=","
              />
            </Typography>

            <Typography color="textSecondary">
              {new Date(lastUpdate).toDateString()}
            </Typography>

            <Typography variant="body2">
              Number of deaths caused by COVID-19
            </Typography>
          </CardContent>
        </Grid>
      </Grid>
    </div>
  );
};

export default Cards;
```



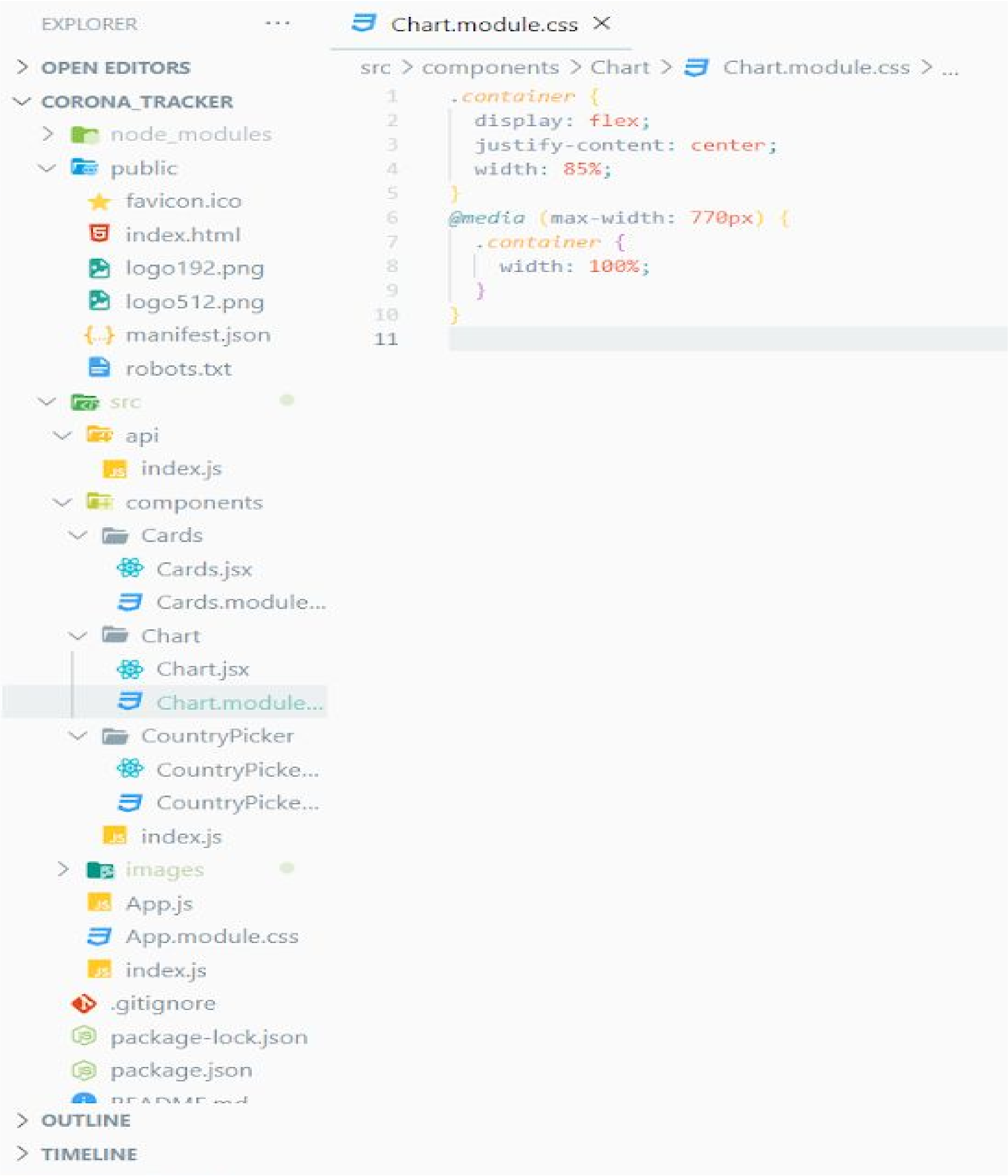
```css
.container {
  margin: 50px 0px;
}
.card {
  margin: 0 2% !important;
}
.infected {
  border-bottom: 10px solid rgba(0, 0, 255, 0.5);
}
.recovered {
  border-bottom: 10px solid rgba(0, 255, 0, 0.5);
}
.deaths {
  border-bottom: 10px solid rgba(255, 0, 0, 0.5);
}

@media (max-width: 770px) {
  .card {
    margin: 2% 0 !important;
  }
}
```



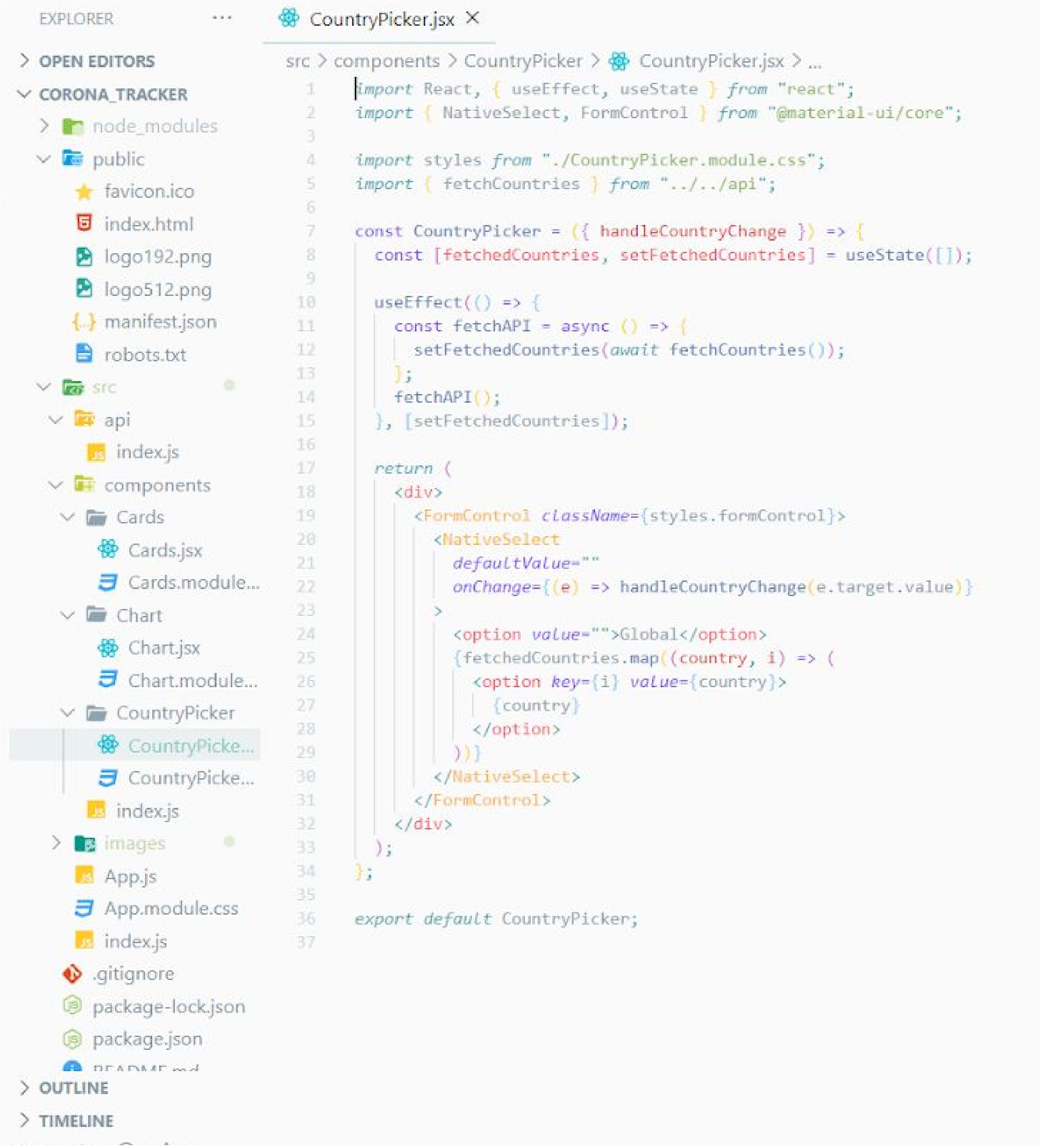
```jsx
import React, { useEffect, useState } from "react";
import { fetchDailyData } from "../../api";
import { Line, Bar } from "react-chartjs-2";
import styles from "./Chart.module.css";

const Chart = ({ data: { confirmed, recovered, deaths }, country }) => {
  const [dailyData, setDailyData] = useState([]);

  useEffect(() => {
    const fetchAPI = async () => {
      setDailyData(await fetchDailyData());
    };

    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;

  const barChart = confirmed ? (
    <Bar
      data={{
        labels: ["Infected", "Recovered", "Deaths"],
        datasets: [
          {
```



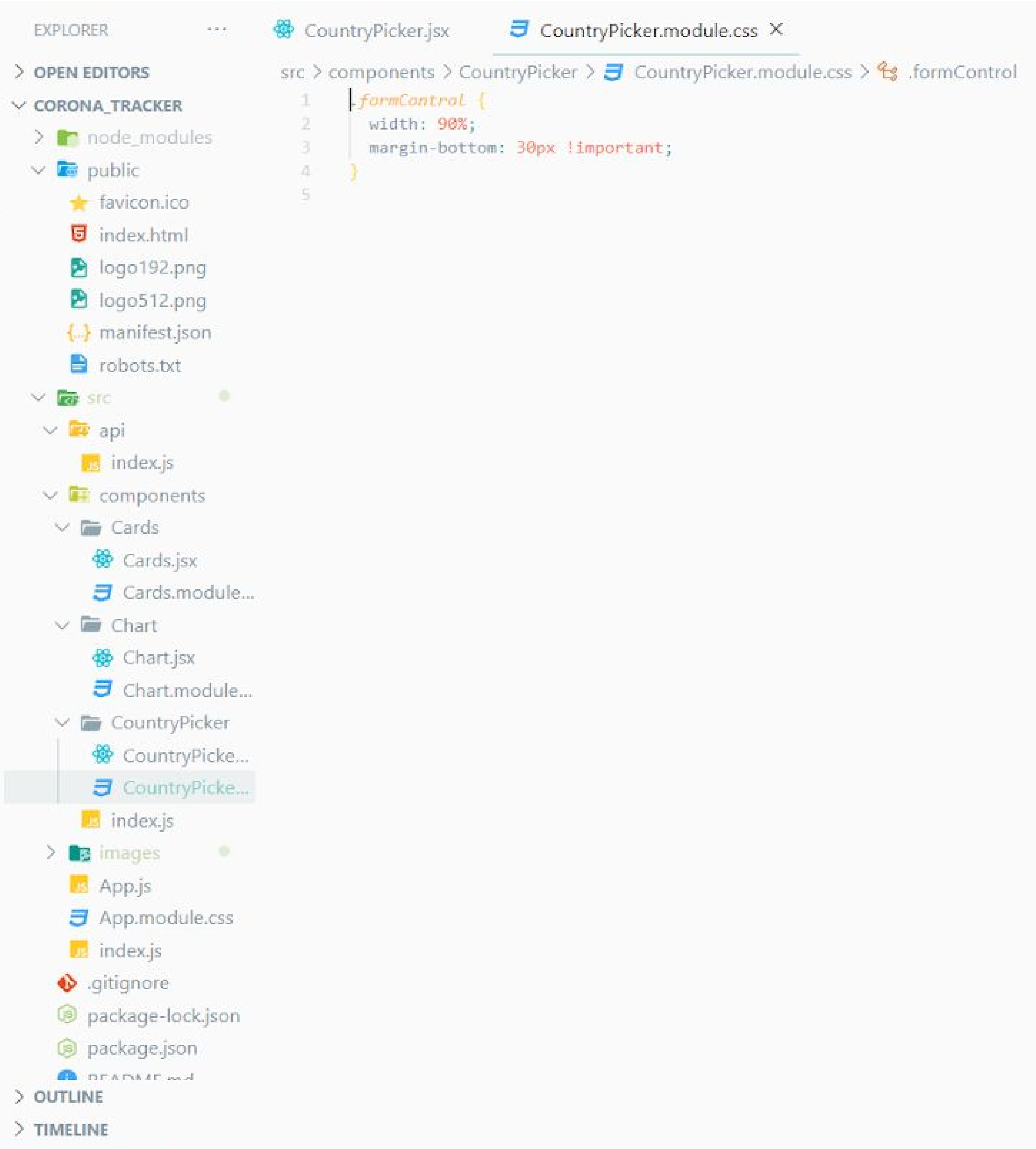
```
  const barChart = confirmed ? (
    <Bar
      data={{
        labels: ["Infected", "Recovered", "Deaths"],
        datasets: [
          {
            label: "People",
            backgroundColor: [
              "rgba(0, 0, 255, 0.5)",
              "rgba(0, 255, 0, 0.5)",
              "rgba(255, 0, 0, 0.5",
            ],
            data: [confirmed.value, recovered.value, deaths.value],
          },
        ],
      }}
      options={{
        legend: { display: false },
        title: { display: true, text: `Current state in ${country}` },
      }}
    />
  ) : null;

  return (
    <div className={styles.container}>{country ? barChart : lineChart}</div>
  );
};

export default Chart;
```



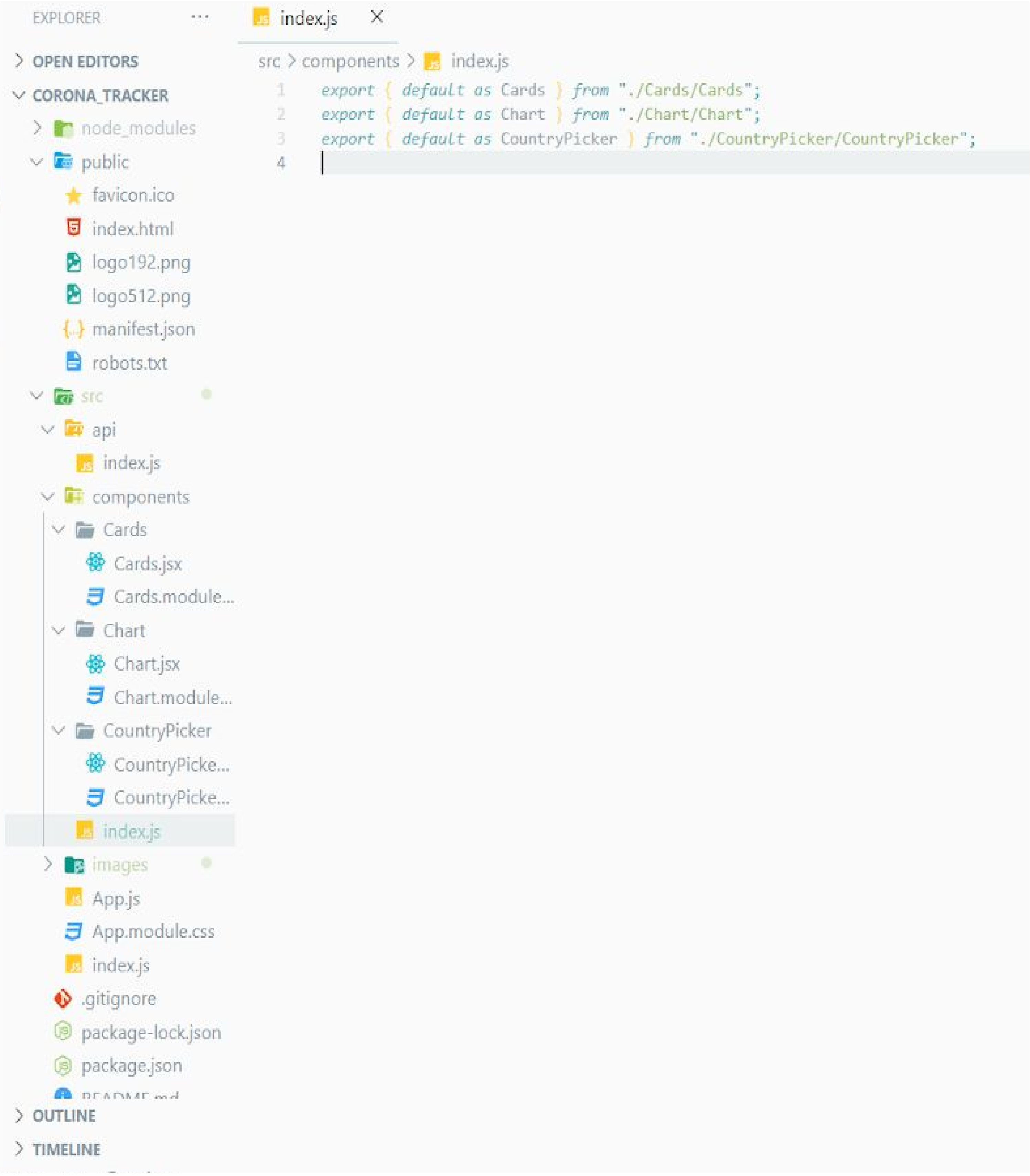
```css
.container {
  display: flex;
  justify-content: center;
  width: 85%;
}
@media (max-width: 770px) {
  .container {
    width: 100%;
  }
}

```



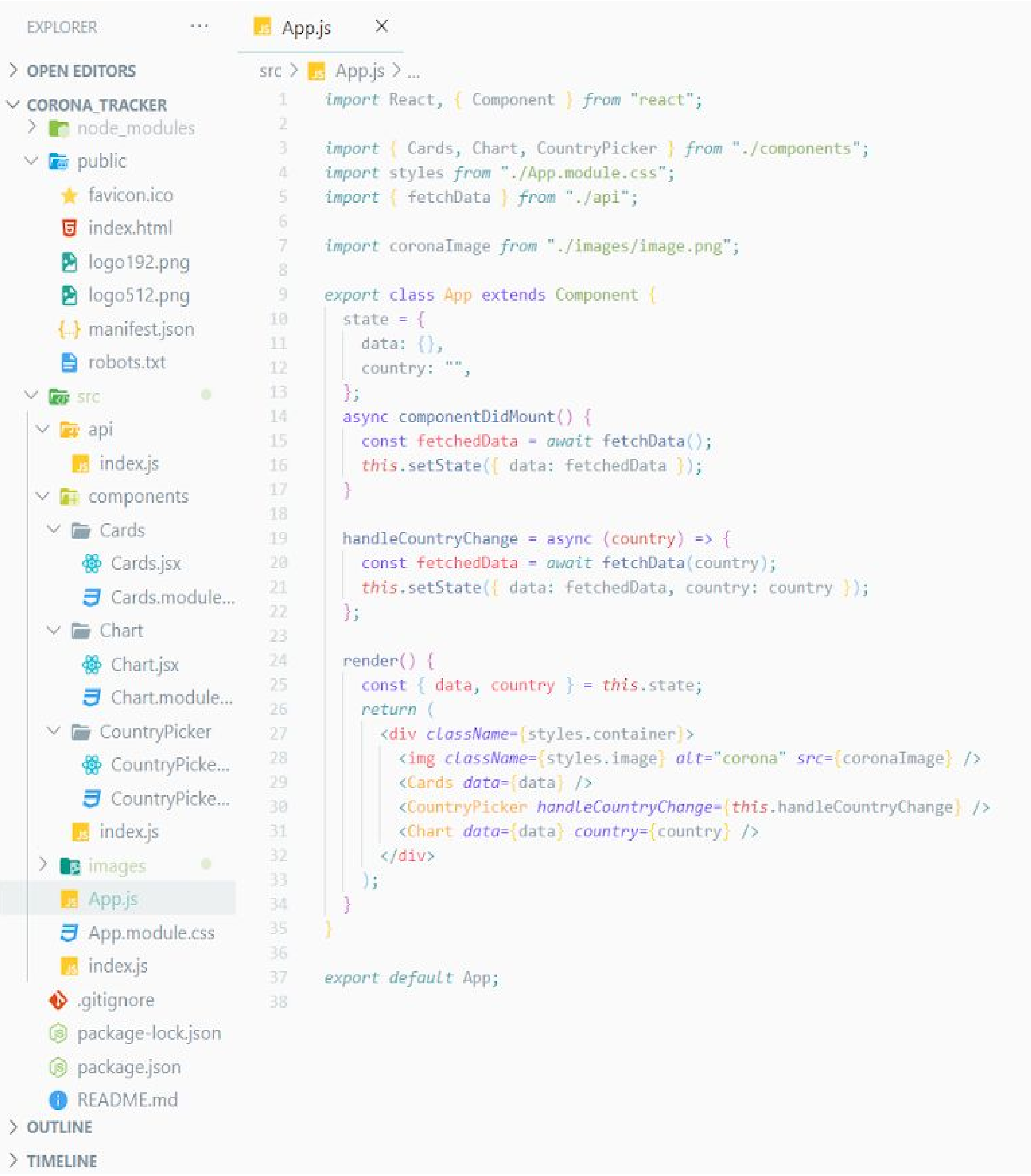
```jsx
import React, { useEffect, useState } from "react";
import { NativeSelect, FormControl } from "@material-ui/core";

import styles from "./CountryPicker.module.css";
import { fetchCountries } from "../../api";

const CountryPicker = ({ handleCountryChange }) => {
  const [fetchedCountries, setFetchedCountries] = useState([]);

  useEffect(() => {
    const fetchAPI = async () => {
      setFetchedCountries(await fetchCountries());
    };
    fetchAPI();
  }, [setFetchedCountries]);

  return (
    <div>
      <FormControl className={styles.formControl}>
        <NativeSelect
          defaultValue=""
          onChange={(e) => handleCountryChange(e.target.value)}
        >
          <option value="">Global</option>
          {fetchedCountries.map((country, i) => (
            <option key={i} value={country}>
              {country}
            </option>
          ))}
        </NativeSelect>
      </FormControl>
    </div>
  );
};

export default CountryPicker;
```



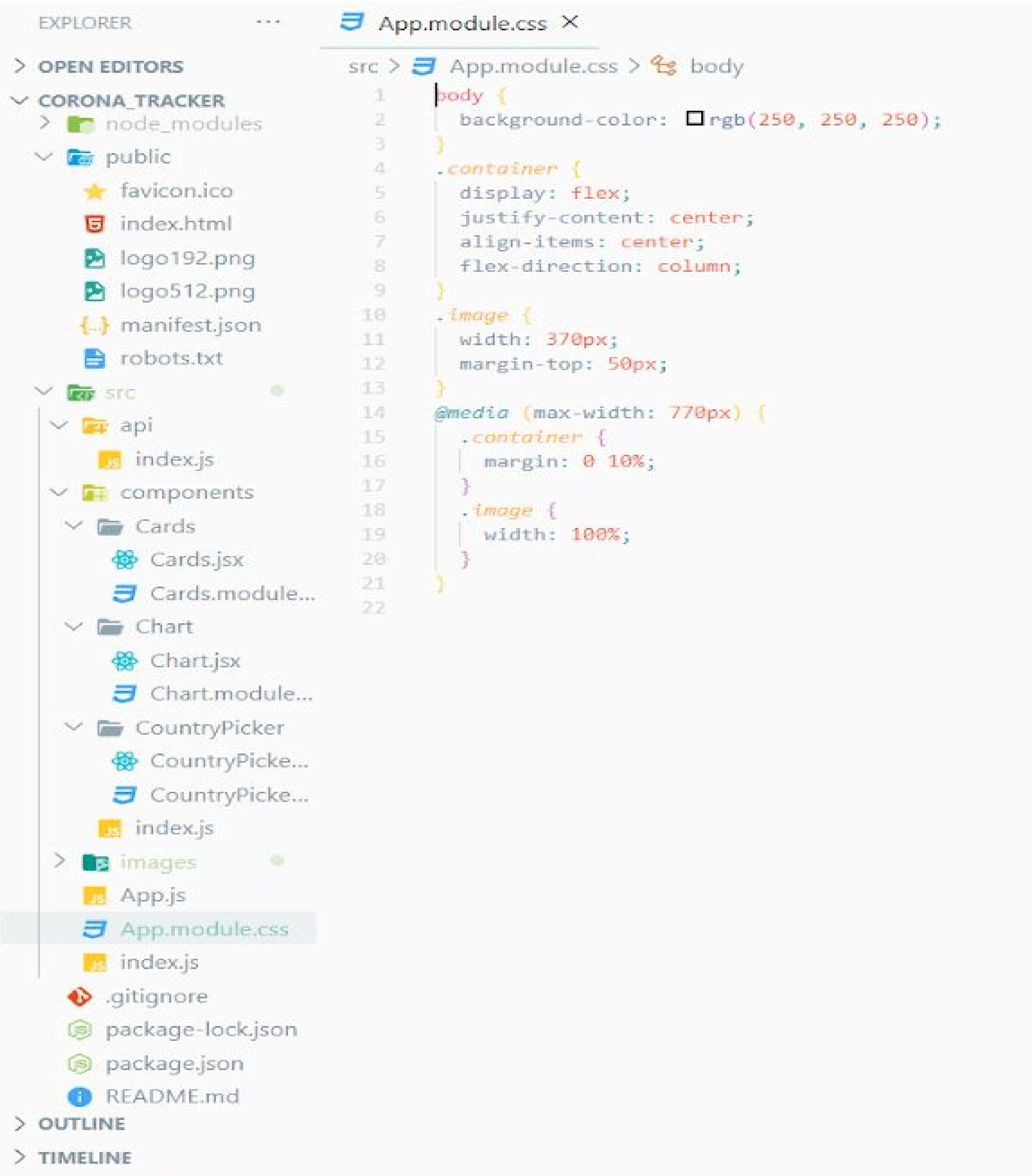
```css
.formControl {
  width: 90%;
  margin-bottom: 30px !important;
}

```



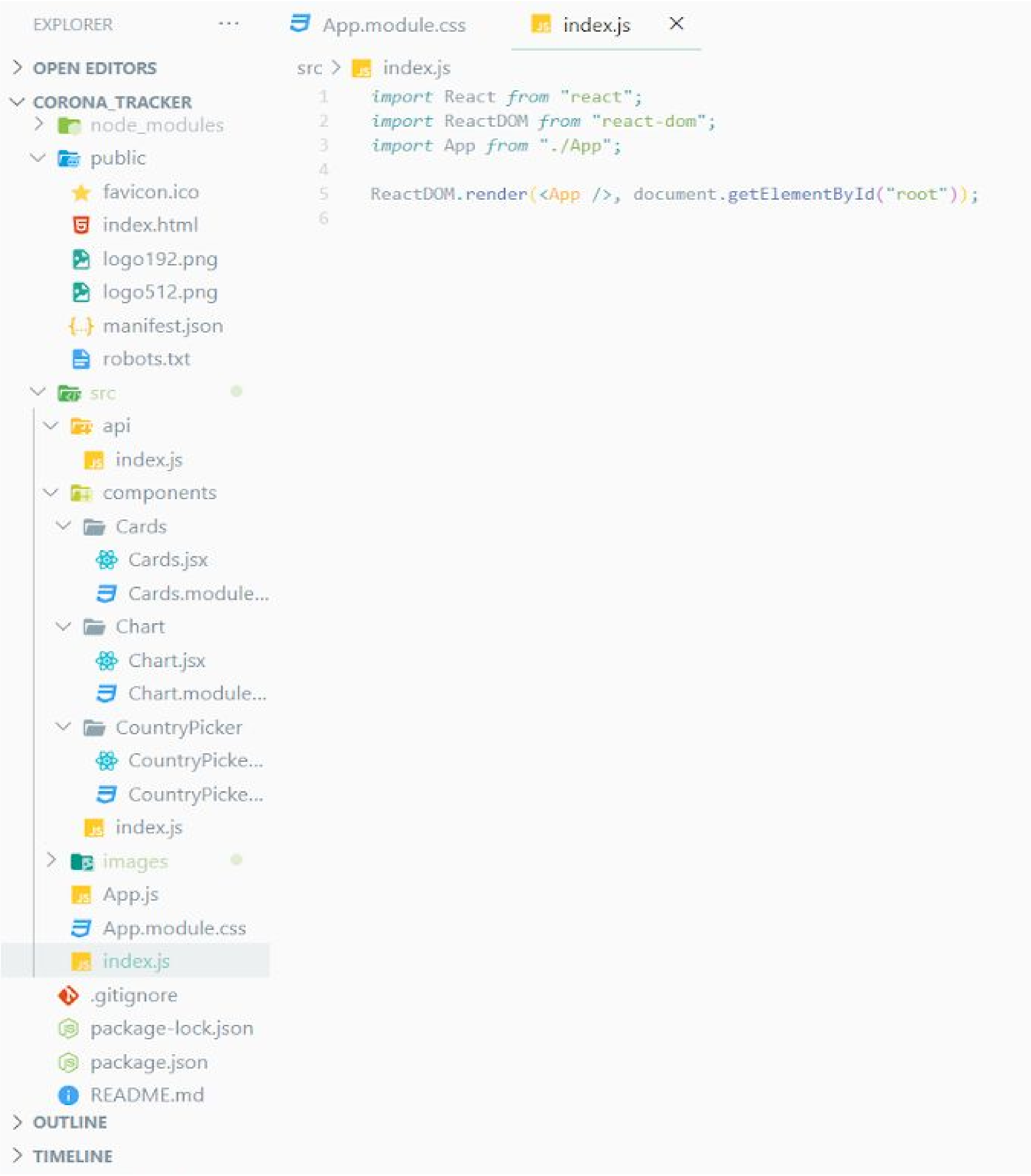
```
export { default as Cards } from "./Cards/Cards";
export { default as Chart } from "./Chart/Chart";
export { default as CountryPicker } from "./CountryPicker/CountryPicker";

```



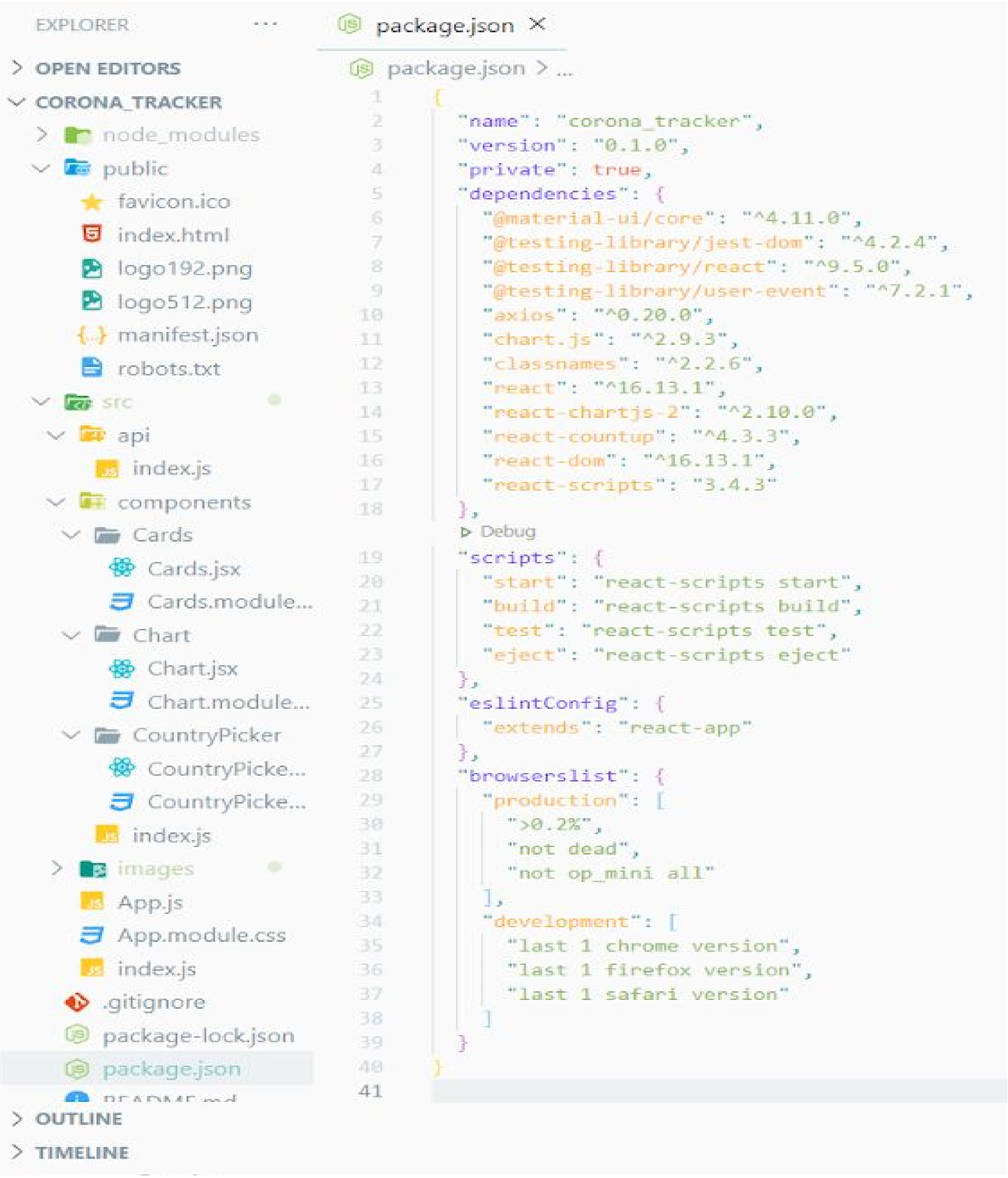
```
import React, { Component } from "react";

import { Cards, Chart, CountryPicker } from "./components";
import styles from "./App.module.css";
import { fetchData } from "./api";

import coronaImage from "./images/image.png";

export class App extends Component {
  state = {
    data: {},
    country: "",
  };
  async componentDidMount() {
    const fetchedData = await fetchData();
    this.setState({ data: fetchedData });
  }

  handleCountryChange = async (country) => {
    const fetchedData = await fetchData(country);
    this.setState({ data: fetchedData, country: country });
  };

  render() {
    const { data, country } = this.state;
    return (
      <div className={styles.container}>
        <img className={styles.image} alt="corona" src={coronaImage} />
        <Cards data={data} />
        <CountryPicker handleCountryChange={this.handleCountryChange} />
        <Chart data={data} country={country} />
      </div>
    );
  }
}

export default App;
```



```css
body {
  background-color: rgb(250, 250, 250);
}
.container {
  display: flex;
  justify-content: center;
  align-items: center;
  flex-direction: column;
}
.image {
  width: 370px;
  margin-top: 50px;
}
@media (max-width: 770px) {
  .container {
    margin: 0 10%;
  }
  .image {
    width: 100%;
  }
}
```



```
import React from "react";
import ReactDOM from "react-dom";
import App from "./App";

ReactDOM.render(<App />, document.getElementById("root"));
```

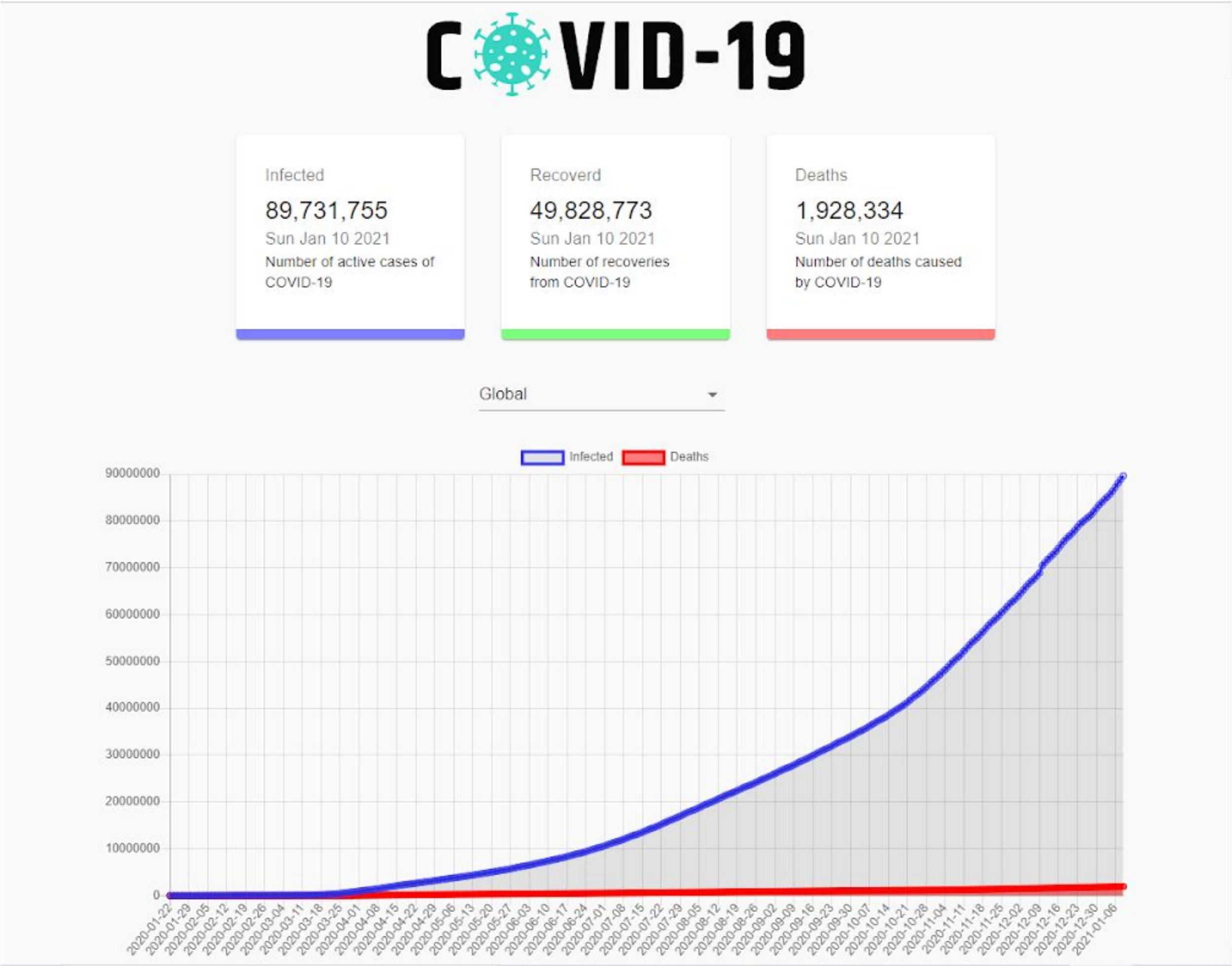


```json
{
  "name": "corona_tracker",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@material-ui/core": "^4.11.0",
    "@testing-library/jest-dom": "^4.2.4",
    "@testing-library/react": "^9.5.0",
    "@testing-library/user-event": "^7.2.1",
    "axios": "^0.20.0",
    "chart.js": "^2.9.3",
    "classnames": "^2.2.6",
    "react": "^16.13.1",
    "react-chartjs-2": "^2.10.0",
    "react-countup": "^4.3.3",
    "react-dom": "^16.13.1",
    "react-scripts": "3.4.3"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": "react-app"
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```

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



C VID-19

Infected
89,731,755
Sun Jan 10 2021
Number of active cases of COVID-19

Recoverd
49,828,773
Sun Jan 10 2021
Number of recoveries from COVID-19

Deaths
1,928,334
Sun Jan 10 2021
Number of deaths caused by COVID-19

Global

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