CRYPTOVERSE(Cryptocurrency-Dashboard)
Nan Mudhalvan Project

Submitted in partial fulfillment of the requirement for the award of

BACHELOR OF COMPUTER SCIENCE

Submitted By

R.MITHRAN-asunm110222201160

S.DINESH-asunm110222201150

E.ANTO CLEMENT-asunm110222201137

P.PRAVEEN-asunm110222201164

UG DEPARTMENT OF COMPUTER SCIENCE

DHARMAMURTHI RAO BAHADUR CALAVALA CUNNAN CHETTY'S

HINDU COLLEGE [SHIFT-II]

"LINGUISTIC (TELUGU) MINORITY STATUS

CONFERRED BY THE GOVT.OF.TAMIL NADU"

Re-Accredited by NAAC –Affillated to the University of Madras

Dharmamurthi Nagar, Pattabiram ,Chennai-600072

DECLARATION

I am **R.MITHRAN**(NM Id:**-asunm110222201160**).Hereby declare that the **Nan Mudhalvan project** work title “**CRYPTOVERSE”** submitted to the UG Department of Computer Science, **SMARTINTERZN**in partial fulfillment of the requirement for the award of Bachelor of Computer Science.

I am **S.DINESH** (NM Id:- **asunm110222201150**).Hereby declare that the **Nan Mudhalvan project** work title “**CRYPTOVERSE”** submitted to the UG Department of Computer Science, **SMARTINTERZN** in partial fulfillment of the requirement for the award of Bachelor of Computer Science.

I am **E.ANTO CLEMENT** (NM Id:- **asunm110222201137**).Hereby declare that the **Nan Mudhalvan project** work title “**CRYPTOVERSE”** submitted to the UG Department of Computer Science, **SMARTINTERZN** in partial fulfillment of the requirement for the award of Bachelor of Computer Science.

I am **P.PRAVEEN**(NM Id:**-asunm110222201164**).Hereby declare that the **Nan Mudhalvan project** work title “**CRYPTOVERSE”** submitted to the UG Department of Computer Science, **SMARTINTERZN** in partial fulfillment of the requirement for the award of Bachelor of Computer Science.

Table Of Content

Cryptoverse

Introduction:

A cryptocurrency dashboard that showcases historical price data over the past five years is an invaluable tool for investors looking to gain a deeper understanding of market trends and dynamics. This feature-rich interface provides a comprehensive historical perspective on the performance of various cryptocurrencies, enabling users to conduct insightful analyses and make informed investment decisions.

With visually intuitive charts and graphs, the dashboard simplifies comparisons between multiple cryptocurrencies, helping users identify top performers and overarching market trends. Customizable timeframes allow for a more detailed examination of price movements, enabling in-depth volatility analysis and risk assessment. By leveraging historical data, investors can recognize recurring patterns and market cycles, enhancing their ability to anticipate potential price movements.

Beyond portfolio optimization, this dashboard also serves as an educational resource, empowering users to understand the evolving nature of cryptocurrency markets and the key factors influencing price fluctuations over time. Whether for seasoned investors or newcomers, this tool provides valuable insights into the long-term behavior of digital assets.

Project overview:

Description

Cryptoverse is a sophisticated cryptocurrency dashboard designed to offer investors deep insights into market dynamics through comprehensive historical price data analysis spanning five years. Equipped with visually intuitive charts, interactive tools, and seamless navigation, the platform enables users to identify top-performing assets and make informed investment decisions.

With its robust search functionality, Cryptoverse allows users to effortlessly explore a wide range of cryptocurrencies and compare their performance over time. Beyond being a powerful tool for portfolio optimization, it also serves as an educational resource, helping users understand the evolving nature of cryptocurrency markets and the key factors influencing price trends.

Key Factors of Cryptoverse

1. **Comprehensive Historical Data** – Provides five years of detailed price history for multiple cryptocurrencies, enabling in-depth market analysis.
2. **Visually Intuitive Charts & Graphs** – Offers dynamic and interactive visualizations for easy comparison of asset performance and trends.
3. **Customizable Timeframes** – Allows users to analyze price movements over different periods for precise trend identification and risk assessment.
4. **Robust Search & Filtering** – Enables quick access to specific cryptocurrencies and efficient comparison across various assets.

Goals of Cryptoverse

1. **Empower Investors** – Provide data-driven insights to help users make informed cryptocurrency investment decisions.
2. **Enhance Market Understanding** – Educate users on cryptocurrency trends, cycles, and factors influencing price movements.
3. **Support Risk Management** – Enable users to assess volatility and make strategic decisions based on historical data.
4. **Facilitate Asset Comparison** – Allow seamless comparison of different cryptocurrencies to identify the best-performing assets.

Pre-requisites:

Here are the key prerequisites for developing a frontend application using

React.js: Node.js and npm:

Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the local environment. It provides a scalable and efficient platform for building network applications.

Install Node.js and npm on your development machine, as they are required to run JavaScript on the server-side.

●Download: https://nodejs.org/en/download/

●Installation instructions: https://nodejs.org/en/download/package-manager/

React.js: React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications.

Installation:

Install React.js, a JavaScript library for building user interfaces.

●Create a newReactapp:

```
npx create-react-app my-react-app
```

Replace my-react-app with your preferred project name.

●Navigate to the project directory:

```
cd my-react-app
```

●Running the React App:

With the React app created, you can now start the development server and see your React application in action.

●Start the development server:

```
npmstart
```

This commandlaunches the development server, and you can access your React app at http://localhost:3000 in your web browser.

HTML,CSS,andJavaScript:Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.

Version Control: Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.

- Git: Download and installation instructions can be found at: https://git-scm.com/downloads

Development Environment: Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.

• Visual Studio Code: Download from : https://code.visualstudio.com/download • Sublime Text: Download from https://www.sublimetext.com/download • WebStorm: Download from https://www.jetbrains.com/webstorm/download

Git repository: https://github.com/SSC369/cryptoverse
Git clone command: git clone

https://github.com/SSC369/cryptoverse Use this command to clone code into your project folder.

Install Dependencies:

- Navigate into the cloned repository directory and install libraries:

cd crypto

Npminstall

Running the Application:

Start the Development Server:

- To start the development server, execute the following command:

npmrundev(vite) or npm start

Access the App:

- Openyour webbrowserandnavigate to http://localhost:3000.

• You should see the Cryptoverse app's homepage, indicating that the installation and setup were successful.

Cryptoverse:

Cryptoverse is an advanced cryptocurrency dashboard designed to provide investors with comprehensive insights into market trends and asset performance using historical price data. By leveraging interactive charts, real-time data visualization, and analytical tools, Cryptoverse empowers users to make informed investment decisions, assess market volatility, and optimize their portfolios effectively.

Core Components of Cryptoverse:

1. Historical Data Analysis

- Provides **five years of historical price data** for multiple cryptocurrencies.
- Helps users identify market trends, recurring patterns, and long-term cycles.
- Enables in-depth analysis of asset performance over customizable timeframes.

2. Interactive Charts & Visualizations

- Offers **dynamic charts and graphs** to display price movements and trends.
- Allows for **custom comparisons** between different cryptocurrencies.
- Supports multiple **chart types** (e.g., line charts, candlestick charts) for diverse analytical needs.

3. Search & Filtering Capabilities

- Users can **easily search and filter** cryptocurrencies based on specific parameters.
- Enables **sorting by market cap, price performance, and volatility** for better decision-making.
- Provides an **intuitive interface** for quick navigation.

4. Customizable Timeframes

- Allows users to **adjust the time range** of historical data analysis.
- Helps in assessing **short-term fluctuations and long-term investment trends**.
- Facilitates a **granular examination of market movements**.

5. Volatility & Risk Assessment Tools

- Offers **historical volatility analysis** to assess investment risks.
- Helps users identify **high-risk and stable assets** for informed portfolio management.
- Supports **risk-reward evaluations** based on past performance trends.

6. Portfolio Optimization & Asset Comparison

- Assists investors in **comparing multiple cryptocurrencies** side by side.
- Highlights **top-performing assets** and market trends to guide investment decisions.
- Supports **diversification strategies** by providing performance insights.

7. Educational Insights & Market Trends

- Acts as a learning resource for both **novice and experienced investors**.
- Explains **market behavior, price fluctuations, and influencing factors**.
- Offers **insightful articles, analysis reports, and trend forecasts**.

8. Seamless User Experience

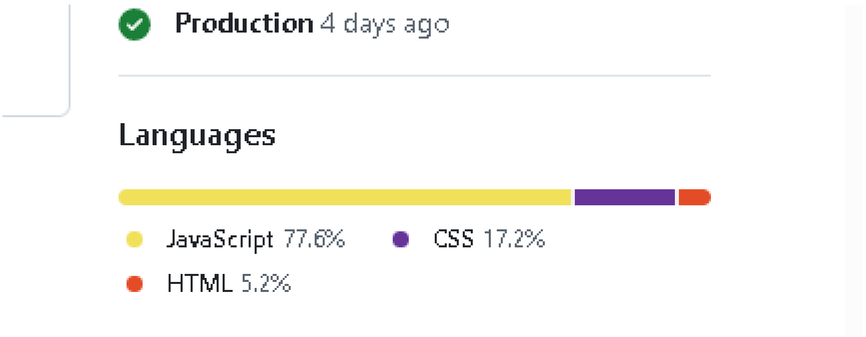
- Features a **clean and intuitive UI** for smooth navigation.
- Ensures **fast data processing** for real-time responsiveness.
- Provides **mobile-friendly accessibility** for on-the-go analysis.

Languages used:

Java script :77.6%

CSS:17.2%

HTML:5.2%



Production 4 days ago

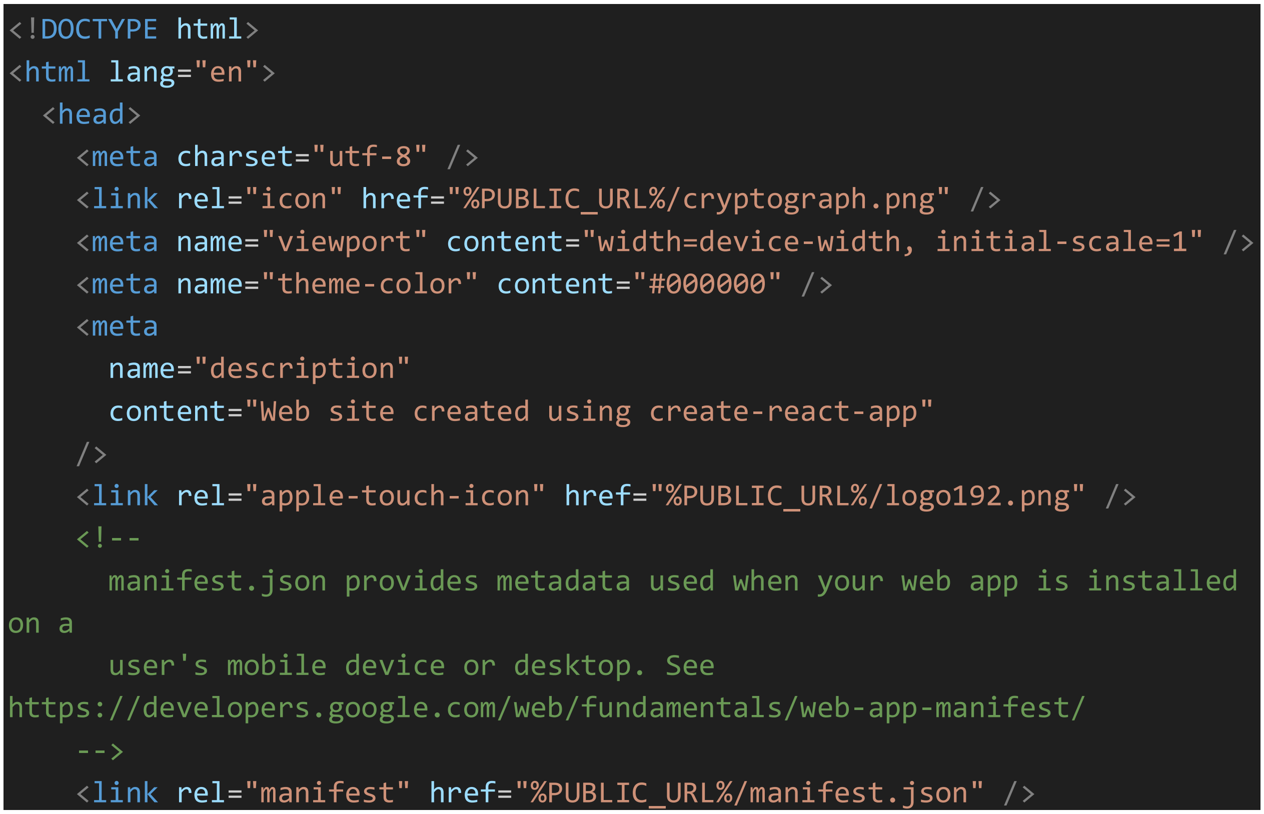
Languages

● JavaScript 77.6%   ● CSS 17.2%

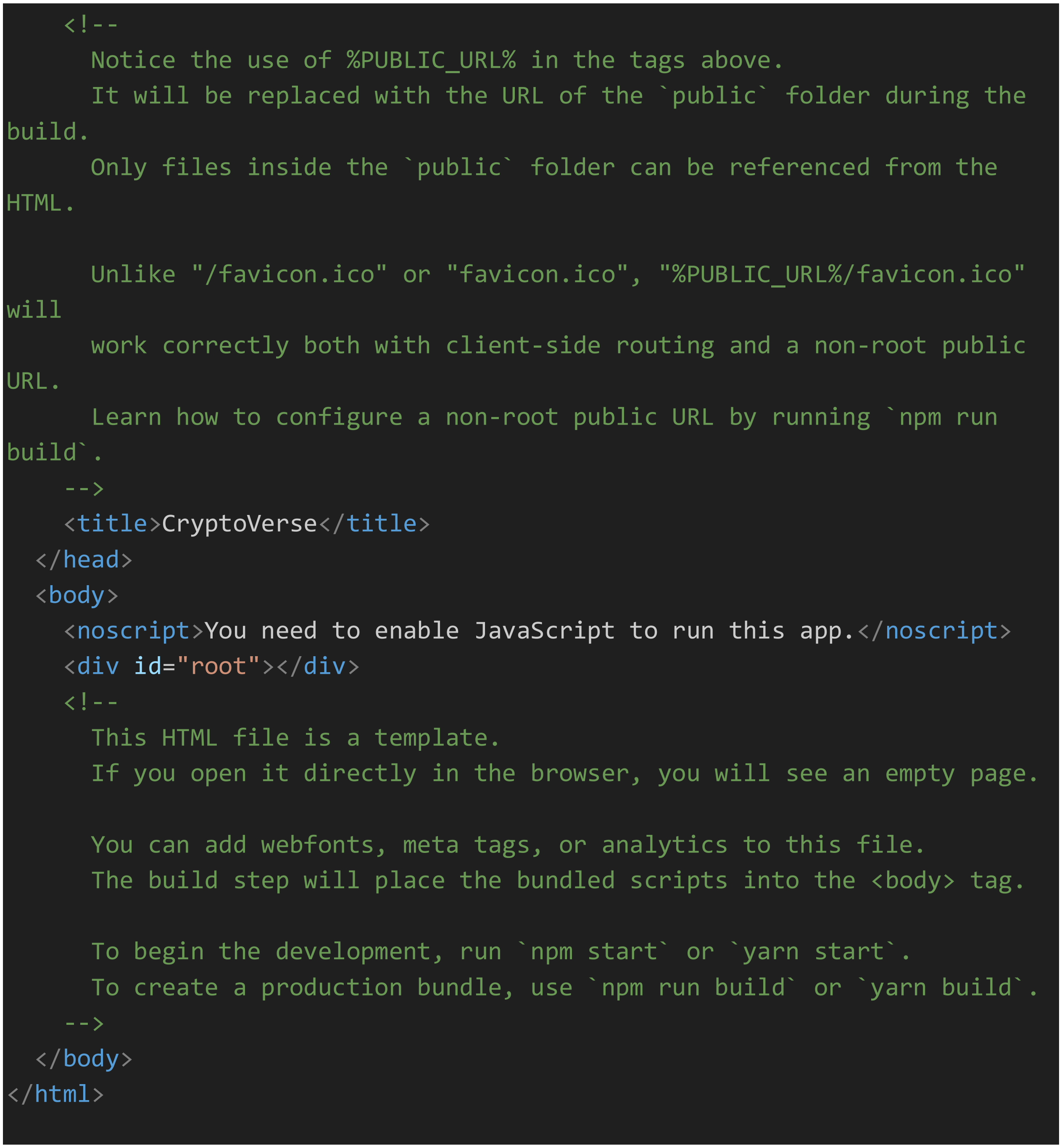
● HTML 5.2%

Coding:

Html:



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/cryptograph.png" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <!--
      manifest.json provides metadata used when your web app is installed
on a
      user's mobile device or desktop. See
https://developers.google.com/web/fundamentals/web-app-manifest/
    -->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
```



```
    <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder during the
build.
      Only files inside the `public` folder can be referenced from the
HTML.

      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico"
will
      work correctly both with client-side routing and a non-root public
URL.
      Learn how to configure a non-root public URL by running `npm run
build`.
    -->
    <title>CryptoVerse</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
    <!--
      This HTML file is a template.
      If you open it directly in the browser, you will see an empty page.

      You can add webfonts, meta tags, or analytics to this file.
      The build step will place the bundled scripts into the <body> tag.

      To begin the development, run `npm start` or `yarn start`.
      To create a production bundle, use `npm run build` or `yarn build`.
    -->
  </body>
</html>
```

Manifest.json:

```json
{
  "short_name": "React App",
  "name": "Create React App Sample",
  "icons": [
    {
      "src": "favicon.ico",
      "sizes": "64x64 32x32 24x24 16x16",
      "type": "image/x-icon"
    },
    {
      "src": "logo192.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "logo512.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ],
  "start_url": ".",
  "display": "standalone",
  "theme_color": "#000000",
  "background_color": "#ffffff"
}
```

App:

Store.js:

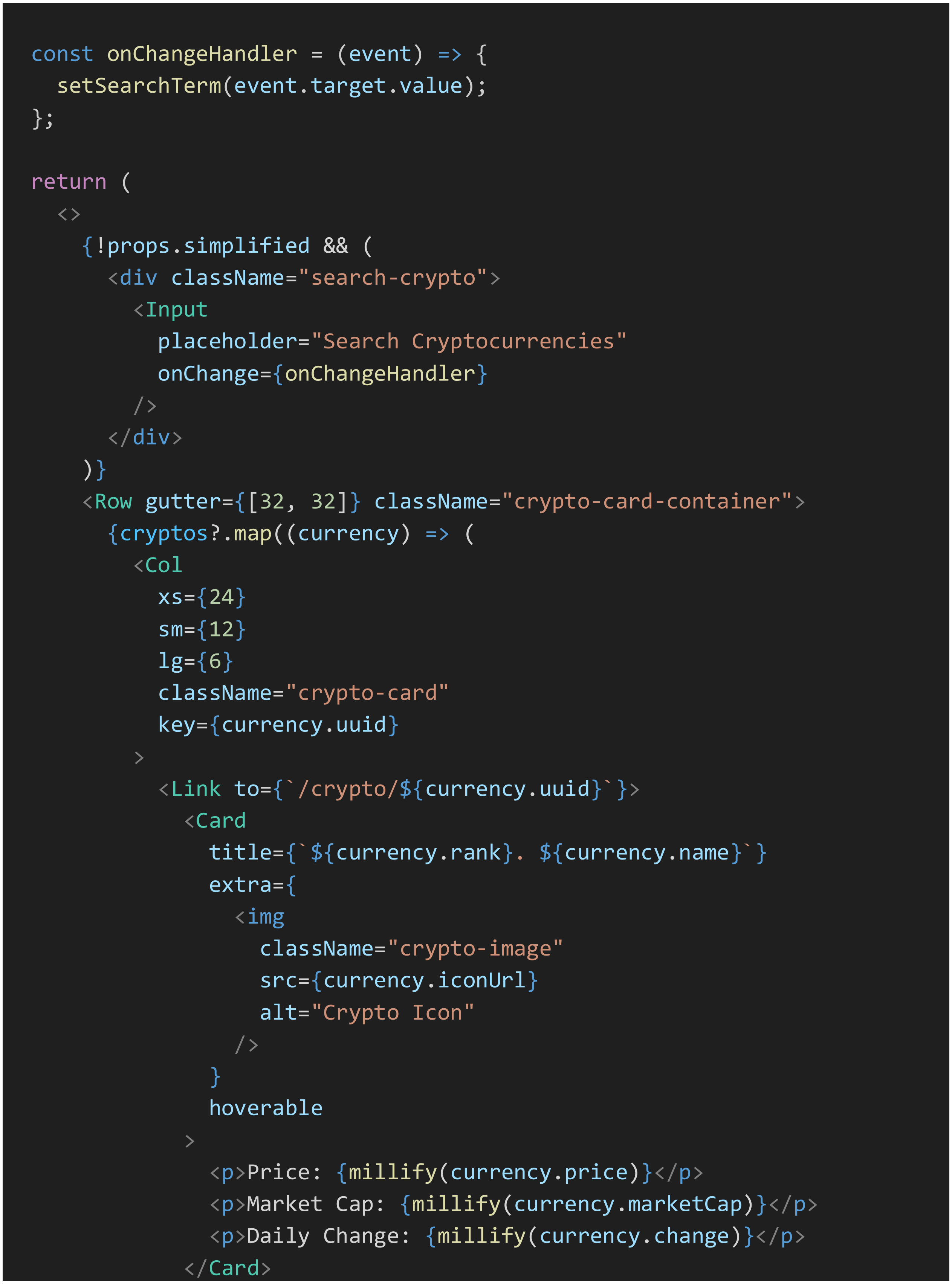
```javascript
import { configureStore } from "@reduxjs/toolkit";
import { cryptoApi } from "../services/cryptoApi";
import { cryptoNewsApi } from "../services/cryptoNewsApi";
```

```
const store = configureStore({
  reducer: {
    [cryptoApi.reducerPath]: cryptoApi.reducer,
    [cryptoNewsApi.reducerPath]: cryptoNewsApi.reducer,
  },
});

export default store;
```

Components of js:

Cryptocurrency:

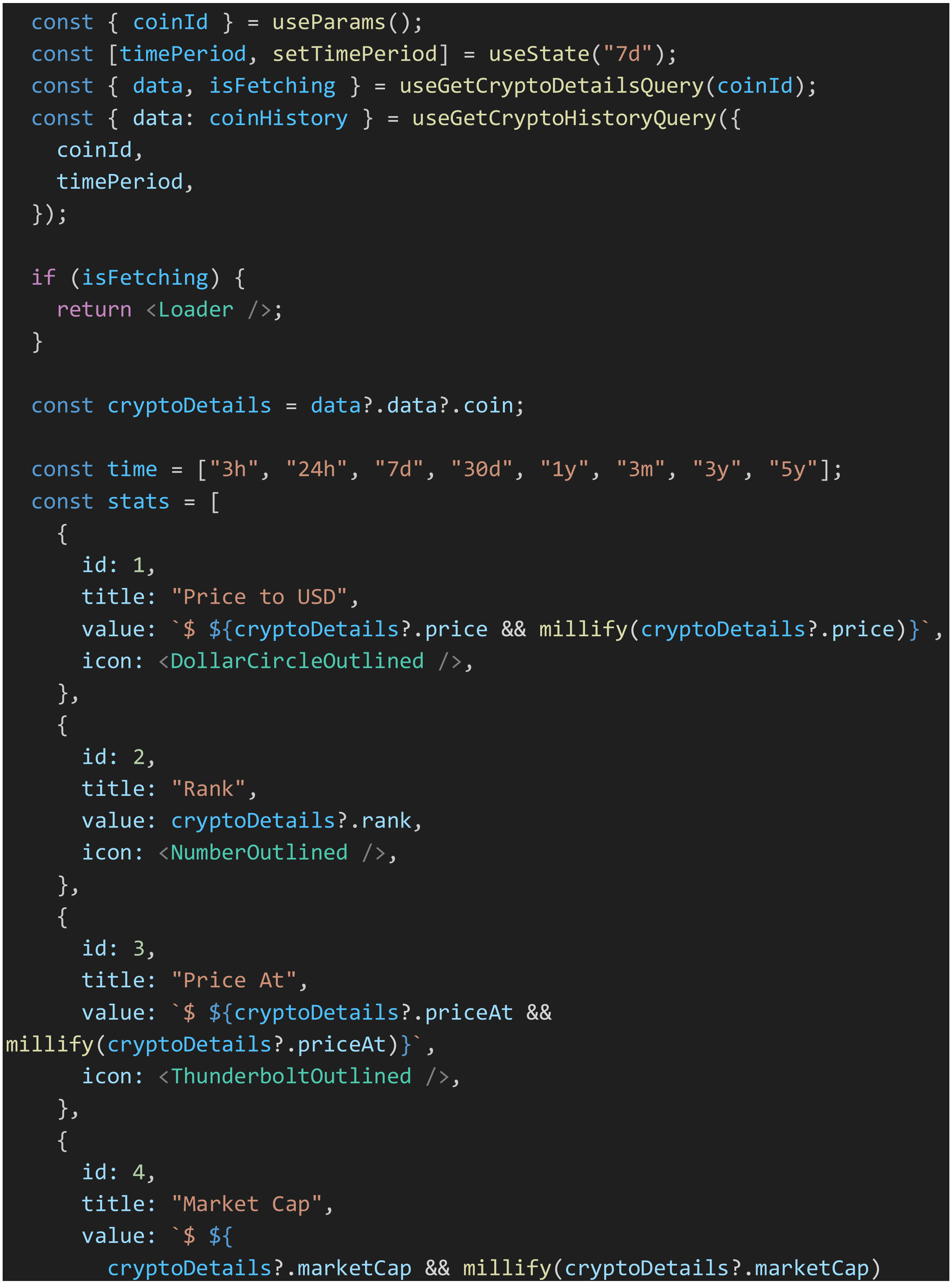
```
import React, { useState, useEffect } from "react";
import millify from "millify";
import { Link } from "react-router-dom";
import { Card, Row, Col, Input } from "antd";
import Loader from "./Loader";
import { useGetCryptosQuery } from "../services/cryptoApi";

const Cryptocurrency = (props) => {
  const count = props.simplified ? 10 : 100;
  const { data: cryptosList, isFetching } = useGetCryptosQuery(count);
  const [cryptos, setCryptos] = useState([]);
  const [searchTerm, setSearchTerm] = useState("");

  useEffect(() => {
    const filteredData = cryptosList?.data?.coins.filter((coin) =>
      coin.name.toLowerCase().includes(searchTerm.toLowerCase())
    );

    setCryptos(filteredData);
  }, [searchTerm, cryptosList]);

  if (isFetching) {
    return <Loader />;
  }
```



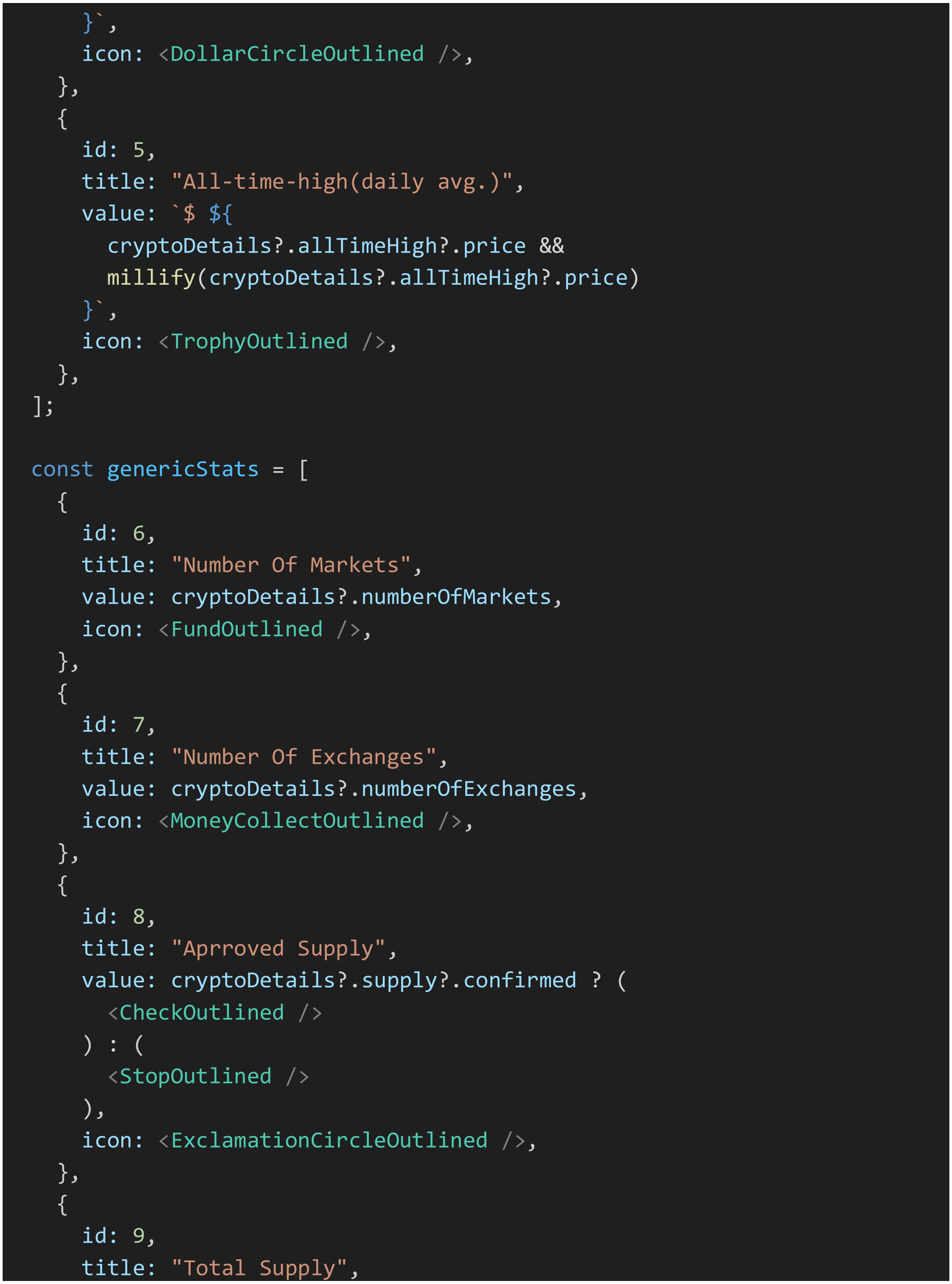
```
const onChangeHandler = (event) => {
  setSearchTerm(event.target.value);
};

return (
  <>
    {!props.simplified && (
      <div className="search-crypto">
        <Input
          placeholder="Search Cryptocurrencies"
          onChange={onChangeHandler}
        />
      </div>
    )}
    <Row gutter={[32, 32]} className="crypto-card-container">
      {cryptos?.map((currency) => (
        <Col
          xs={24}
          sm={12}
          lg={6}
          className="crypto-card"
          key={currency.uuid}
        >
          <Link to={`/crypto/${currency.uuid}`}>
            <Card
              title={`${currency.rank}. ${currency.name}`}
              extra={
                <img
                  className="crypto-image"
                  src={currency.iconUrl}
                  alt="Crypto Icon"
                />
              }
              hoverable
            >
              <p>Price: {millify(currency.price)}</p>
              <p>Market Cap: {millify(currency.marketCap)}</p>
              <p>Daily Change: {millify(currency.change)}</p>
            </Card>
            </Link>
          </Col>
        ))}
      </Row>
    </>
  );
};

export default Cryptocurrency;
```

cryptodetails:

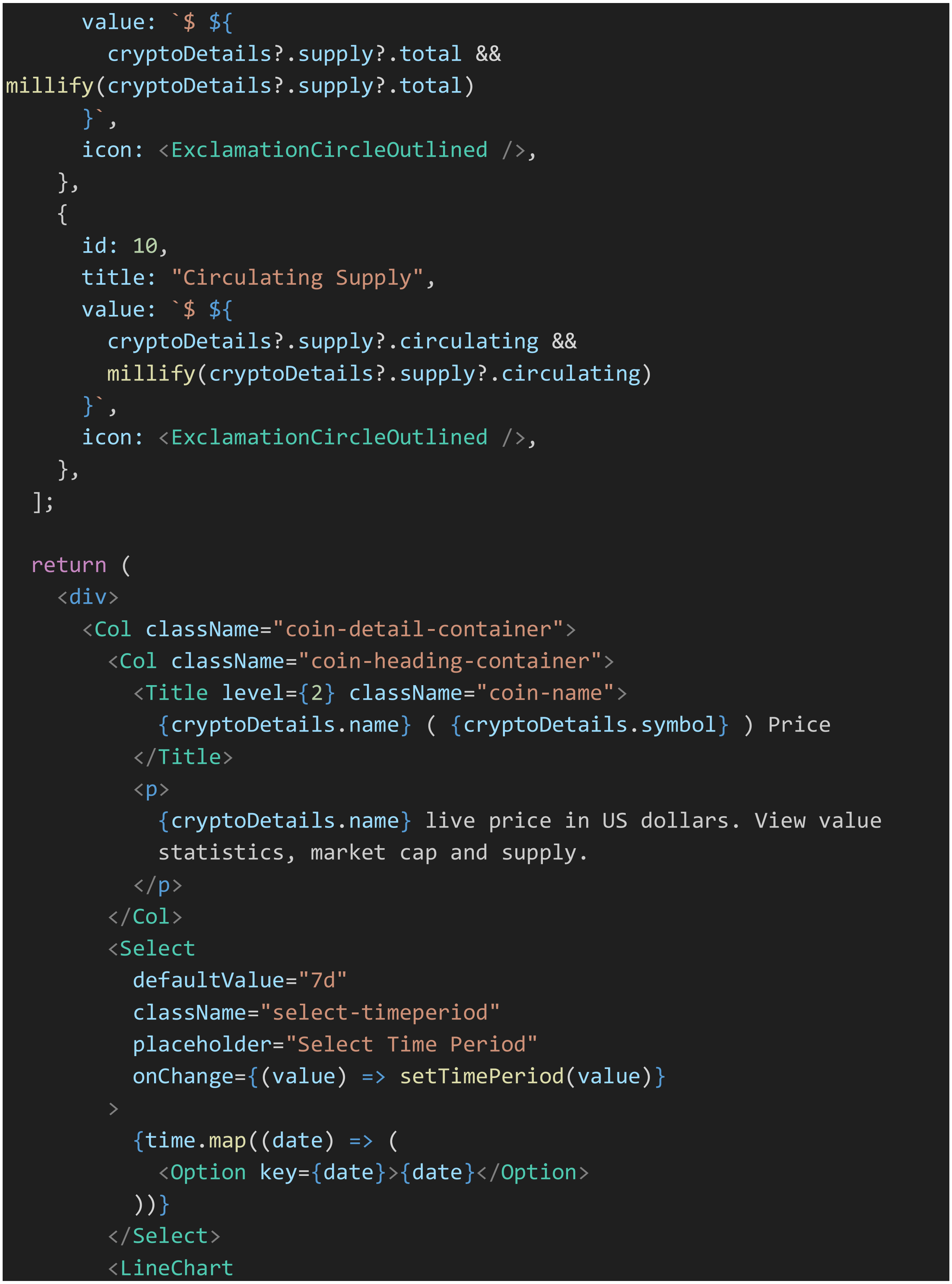
```
import React, { useState } from "react";
import HTMLReactParser from "html-react-parser";
import { useParams } from "react-router-dom";
import millify from "millify";
import { Col, Row, Typography, Select } from "antd";
import {
  MoneyCollectOutlined,
  DollarCircleOutlined,
  FundOutlined,
  ExclamationCircleOutlined,
  StopOutlined,
  TrophyOutlined,
  CheckOutlined,
  NumberOutlined,
  ThunderboltOutlined,
} from "@ant-design/icons";
import {
  useGetCryptoDetailsQuery,
  useGetCryptoHistoryQuery,
} from "../services/cryptoApi";
import LineChart from "./LineChart";
import Loader from "./Loader";

const { Title, Text } = Typography;
const { Option } = Select;

const CryptoDetails = () => {
```



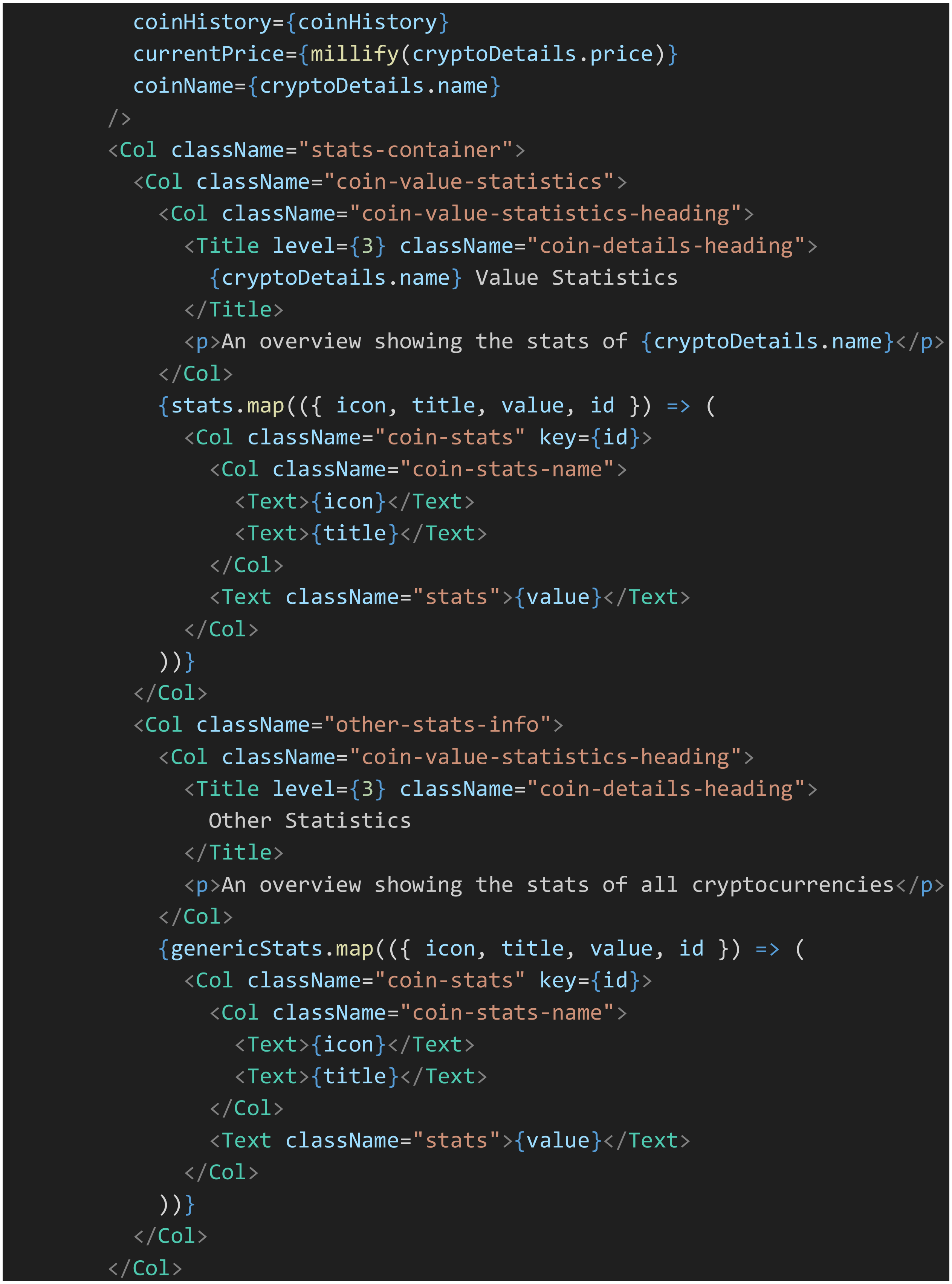
```
  const { coinId } = useParams();
  const [timePeriod, setTimePeriod] = useState("7d");
  const { data, isFetching } = useGetCryptoDetailsQuery(coinId);
  const { data: coinHistory } = useGetCryptoHistoryQuery({
    coinId,
    timePeriod,
  });

  if (isFetching) {
    return <Loader />;
  }

  const cryptoDetails = data?.data?.coin;

  const time = ["3h", "24h", "7d", "30d", "1y", "3m", "3y", "5y"];
  const stats = [
    {
      id: 1,
      title: "Price to USD",
      value: `$ ${cryptoDetails?.price && millify(cryptoDetails?.price)}`,
      icon: <DollarCircleOutlined />,
    },
    {
      id: 2,
      title: "Rank",
      value: cryptoDetails?.rank,
      icon: <NumberOutlined />,
    },
    {
      id: 3,
      title: "Price At",
      value: `$ ${cryptoDetails?.priceAt &&
millify(cryptoDetails?.priceAt)}`,
      icon: <ThunderboltOutlined />,
    },
    {
      id: 4,
      title: "Market Cap",
      value: `$ ${
        cryptoDetails?.marketCap && millify(cryptoDetails?.marketCap)
```



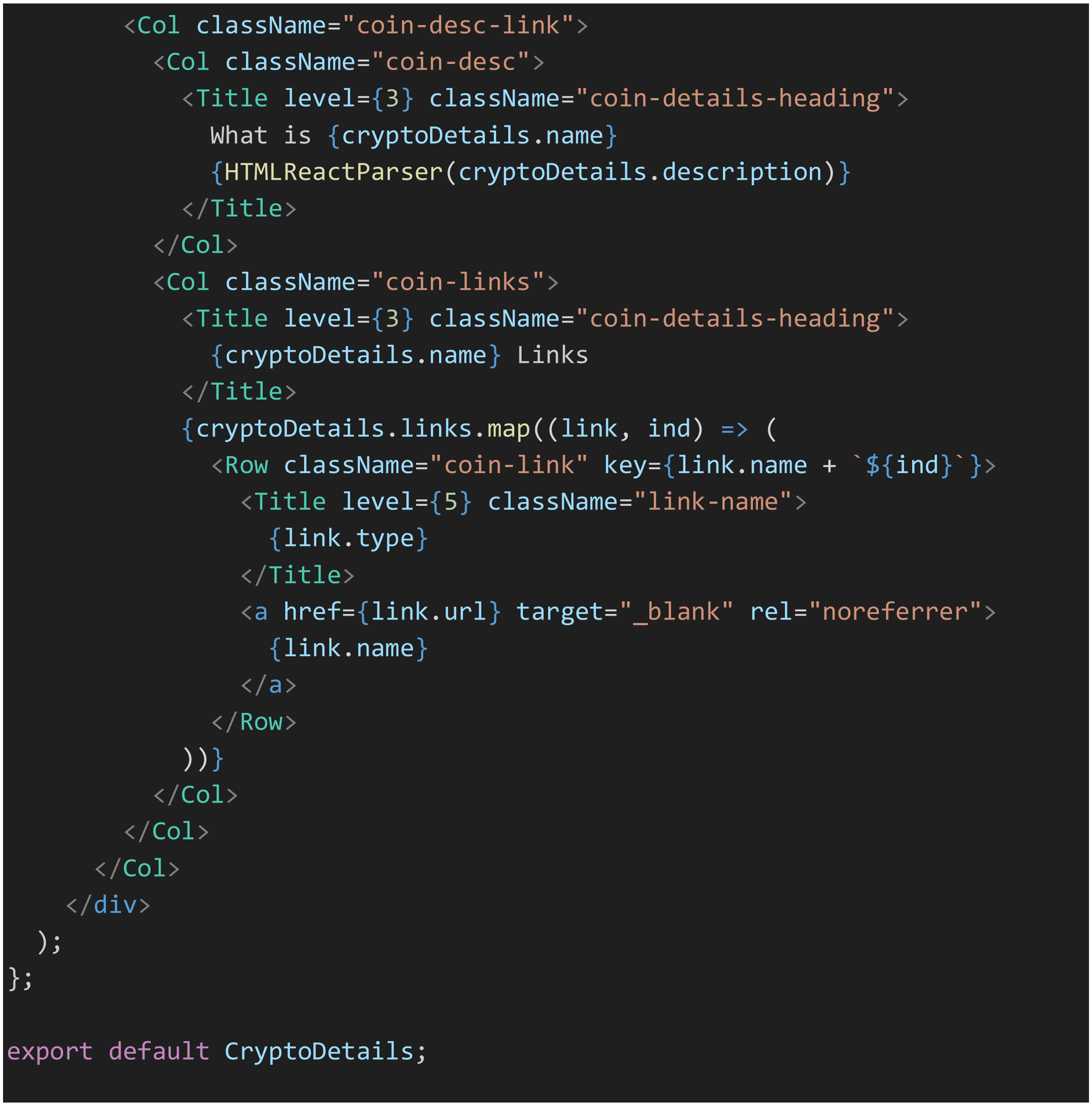
```
    }`,
    icon: <DollarCircleOutlined />,
  },
  {
    id: 5,
    title: "All-time-high(daily avg.)",
    value: `$ ${
      cryptoDetails?.allTimeHigh?.price &&
      millify(cryptoDetails?.allTimeHigh?.price)
    }`,
    icon: <TrophyOutlined />,
  },
];

const genericStats = [
  {
    id: 6,
    title: "Number Of Markets",
    value: cryptoDetails?.numberOfMarkets,
    icon: <FundOutlined />,
  },
  {
    id: 7,
    title: "Number Of Exchanges",
    value: cryptoDetails?.numberOfExchanges,
    icon: <MoneyCollectOutlined />,
  },
  {
    id: 8,
    title: "Aprroved Supply",
    value: cryptoDetails?.supply?.confirmed ? (
      <CheckOutlined />
    ) : (
      <StopOutlined />
    ),
    icon: <ExclamationCircleOutlined />,
  },
  {
    id: 9,
    title: "Total Supply",
```



```
      value: `$ ${
        cryptoDetails?.supply?.total &&
millify(cryptoDetails?.supply?.total)
      }`,
      icon: <ExclamationCircleOutlined />,
    },
    {
      id: 10,
      title: "Circulating Supply",
      value: `$ ${
        cryptoDetails?.supply?.circulating &&
        millify(cryptoDetails?.supply?.circulating)
      }`,
      icon: <ExclamationCircleOutlined />,
    },
  ];

  return (
    <div>
      <Col className="coin-detail-container">
        <Col className="coin-heading-container">
          <Title level={2} className="coin-name">
            {cryptoDetails.name} ( {cryptoDetails.symbol} ) Price
          </Title>
          <p>
            {cryptoDetails.name} live price in US dollars. View value
            statistics, market cap and supply.
          </p>
        </Col>
        <Select
          defaultValue="7d"
          className="select-timeperiod"
          placeholder="Select Time Period"
          onChange={(value) => setTimePeriod(value)}
        >
          {time.map((date) => (
            <Option key={date}>{date}</Option>
          ))}
        </Select>
        <LineChart
```

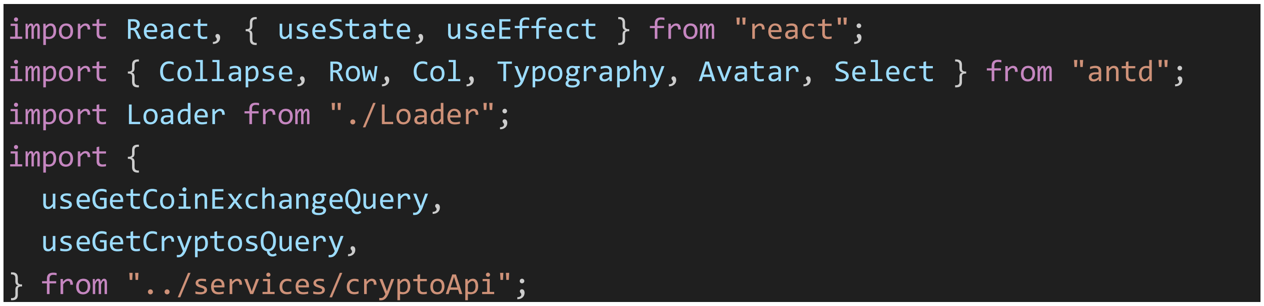


```
          coinHistory={coinHistory}
          currentPrice={millify(cryptoDetails.price)}
          coinName={cryptoDetails.name}
        />
        <Col className="stats-container">
          <Col className="coin-value-statistics">
            <Col className="coin-value-statistics-heading">
              <Title level={3} className="coin-details-heading">
                {cryptoDetails.name} Value Statistics
              </Title>
              <p>An overview showing the stats of {cryptoDetails.name}</p>
            </Col>
            {stats.map(({ icon, title, value, id }) => (
              <Col className="coin-stats" key={id}>
                <Col className="coin-stats-name">
                  <Text>{icon}</Text>
                  <Text>{title}</Text>
                </Col>
                <Text className="stats">{value}</Text>
              </Col>
            ))}
          </Col>
          <Col className="other-stats-info">
            <Col className="coin-value-statistics-heading">
              <Title level={3} className="coin-details-heading">
                Other Statistics
              </Title>
              <p>An overview showing the stats of all cryptocurrencies</p>
            </Col>
            {genericStats.map(({ icon, title, value, id }) => (
              <Col className="coin-stats" key={id}>
                <Col className="coin-stats-name">
                  <Text>{icon}</Text>
                  <Text>{title}</Text>
                </Col>
                <Text className="stats">{value}</Text>
              </Col>
            ))}
          </Col>
        </Col>
```

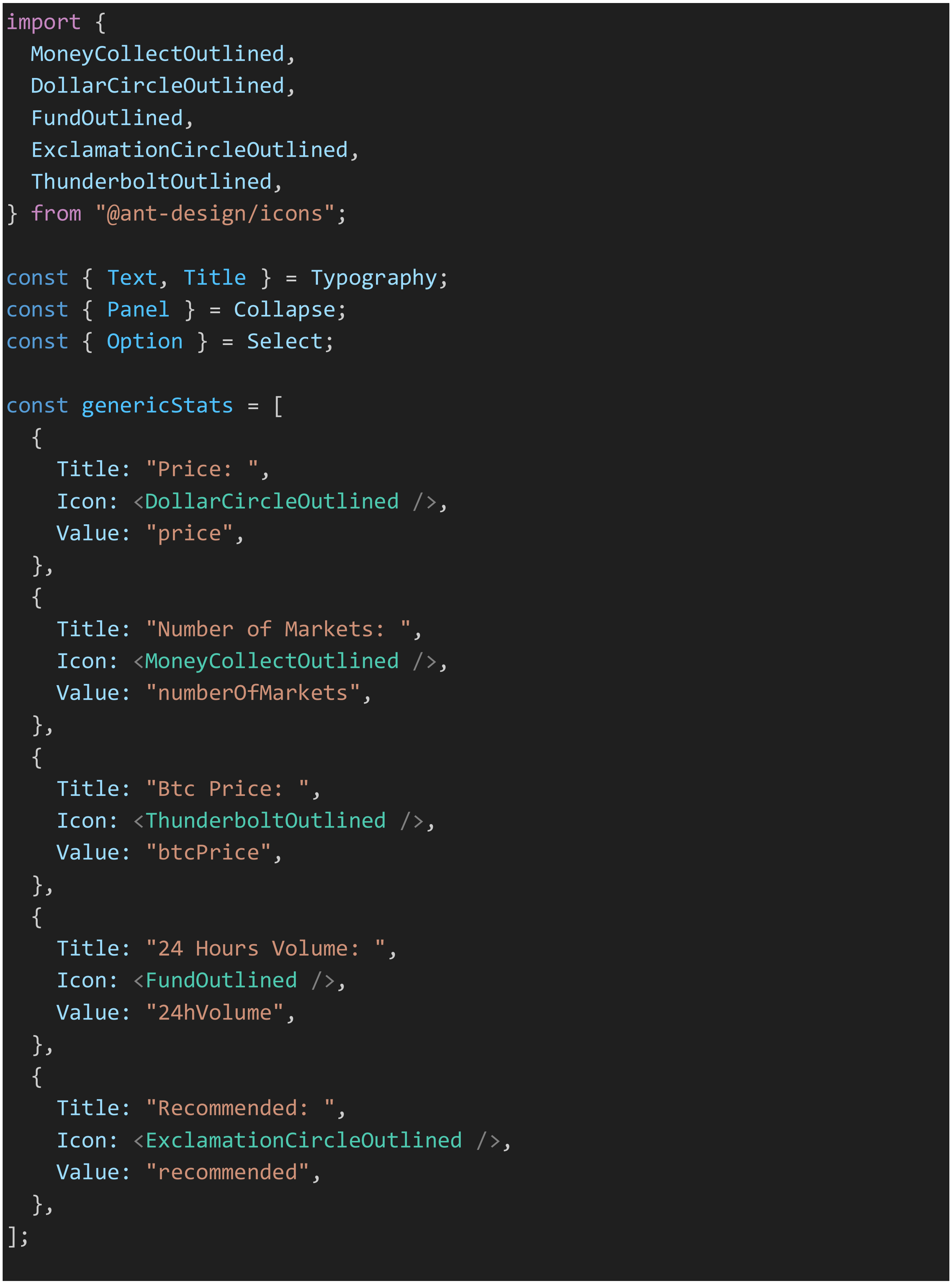


```
        <Col className="coin-desc-link">
          <Col className="coin-desc">
            <Title level={3} className="coin-details-heading">
              What is {cryptoDetails.name}
              {HTMLReactParser(cryptoDetails.description)}
            </Title>
          </Col>
          <Col className="coin-links">
            <Title level={3} className="coin-details-heading">
              {cryptoDetails.name} Links
            </Title>
            {cryptoDetails.links.map((link, ind) => (
              <Row className="coin-link" key={link.name + `${ind}`}>
                <Title level={5} className="link-name">
                  {link.type}
                </Title>
                <a href={link.url} target="_blank" rel="noreferrer">
                  {link.name}
                </a>
              </Row>
            ))}
          </Col>
        </Col>
      </Col>
    </div>
  );
};

export default CryptoDetails;
```

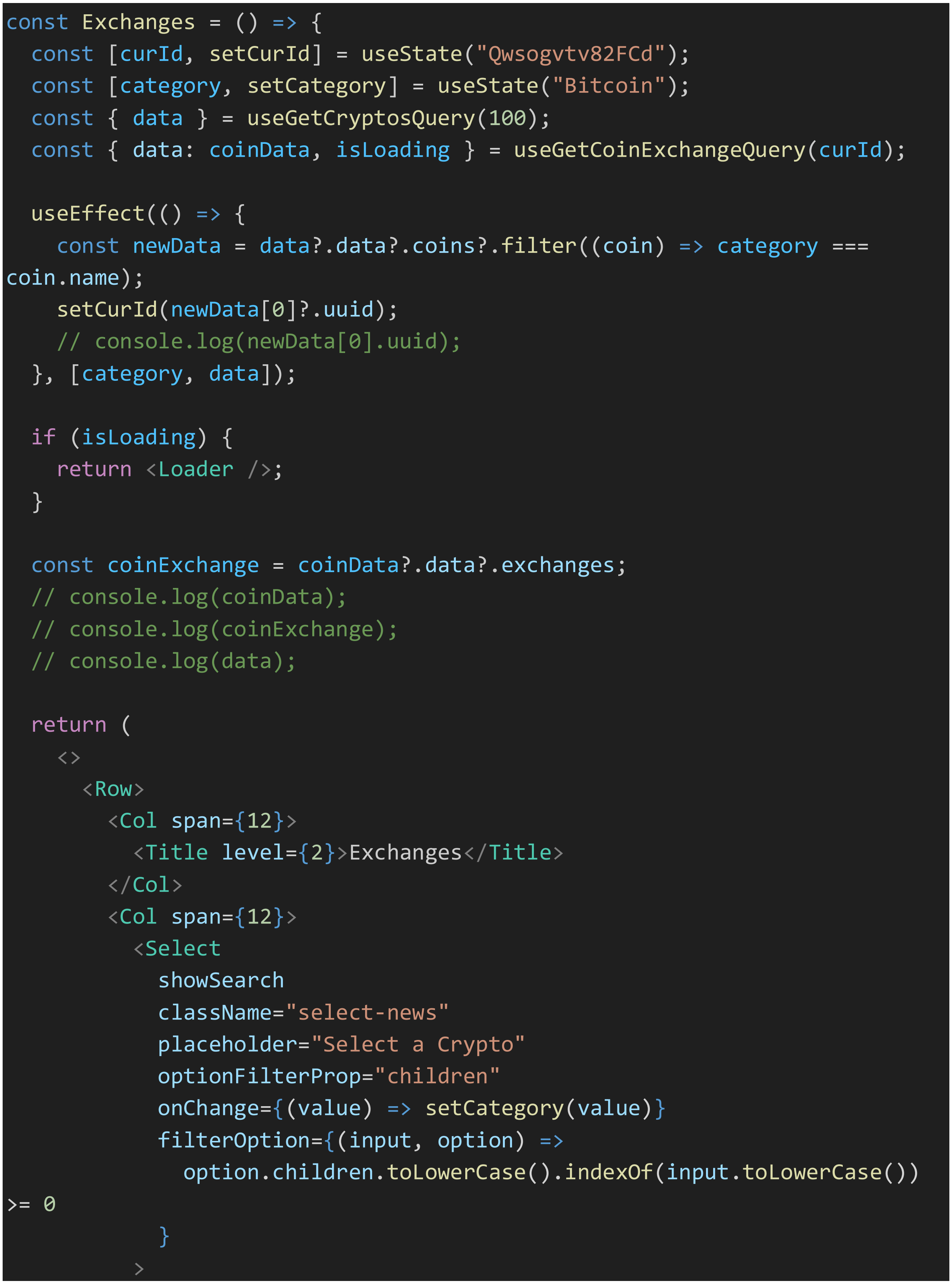
Exchanges:



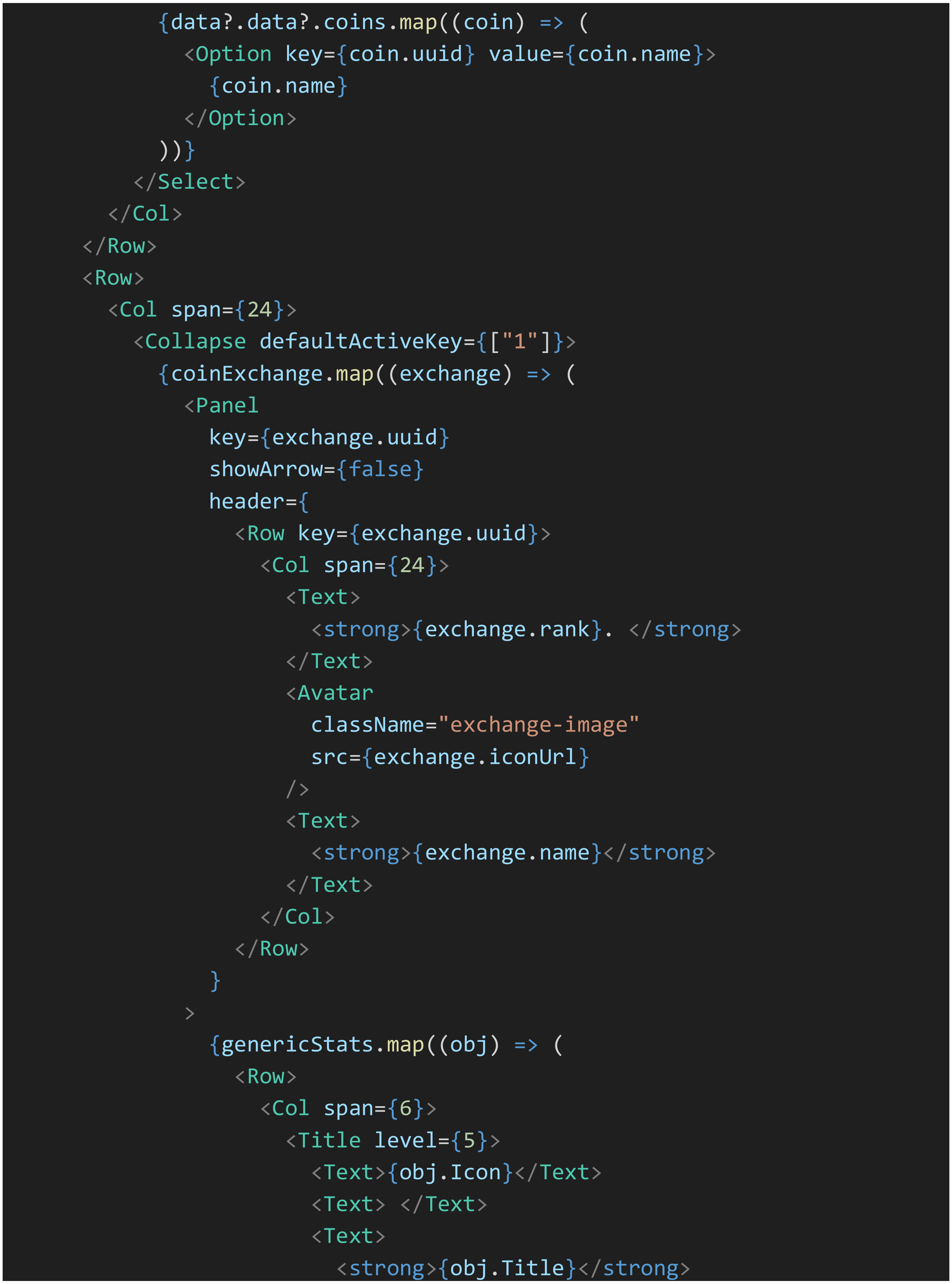
```
import React, { useState, useEffect } from "react";
import { Collapse, Row, Col, Typography, Avatar, Select } from "antd";
import Loader from "./Loader";
import {
  useGetCoinExchangeQuery,
  useGetCryptosQuery,
} from "../services/cryptoApi";
```



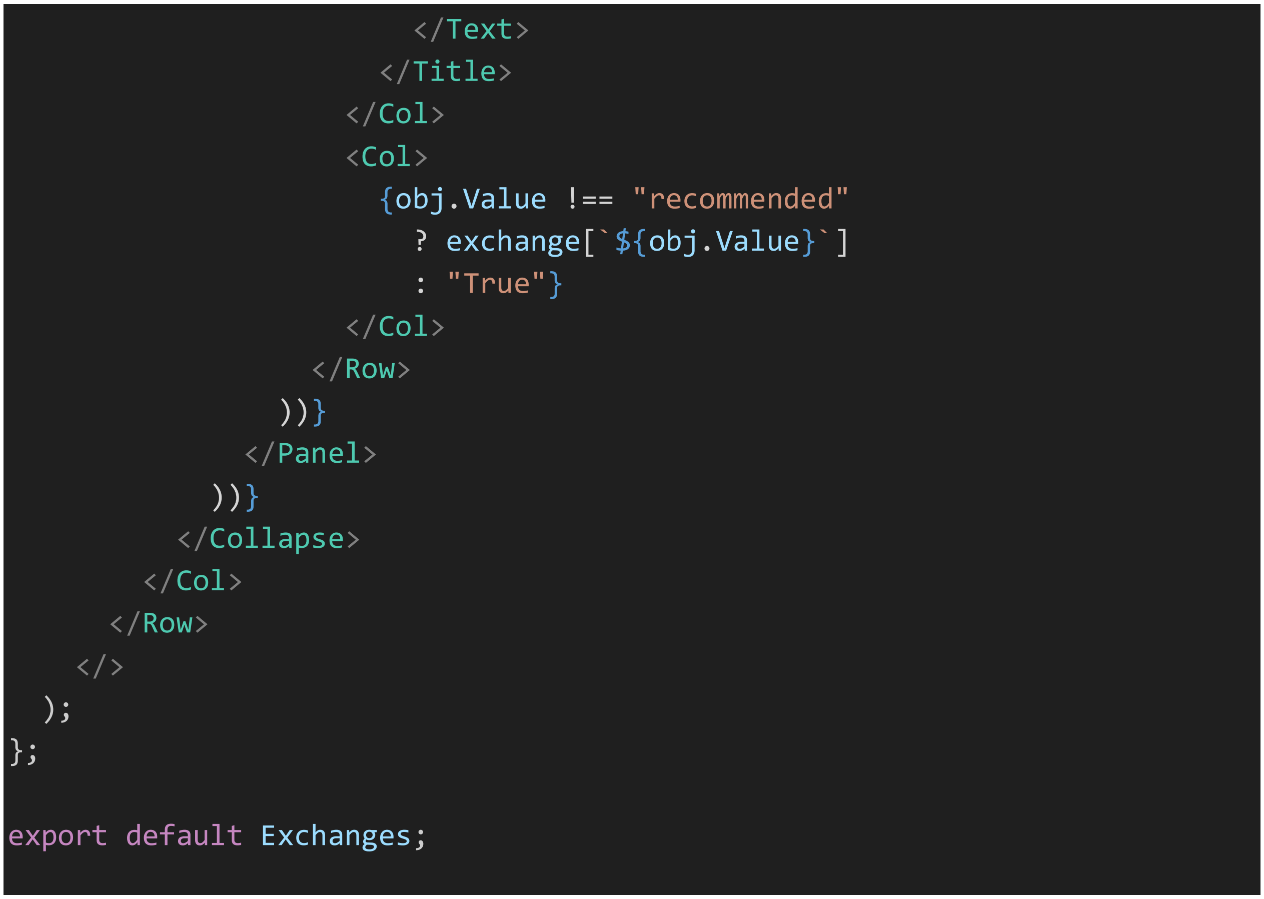
```
import {
  MoneyCollectOutlined,
  DollarCircleOutlined,
  FundOutlined,
  ExclamationCircleOutlined,
  ThunderboltOutlined,
} from "@ant-design/icons";

const { Text, Title } = Typography;
const { Panel } = Collapse;
const { Option } = Select;

const genericStats = [
  {
    Title: "Price: ",
    Icon: <DollarCircleOutlined />,
    Value: "price",
  },
  {
    Title: "Number of Markets: ",
    Icon: <MoneyCollectOutlined />,
    Value: "numberOfMarkets",
  },
  {
    Title: "Btc Price: ",
    Icon: <ThunderboltOutlined />,
    Value: "btcPrice",
  },
  {
    Title: "24 Hours Volume: ",
    Icon: <FundOutlined />,
    Value: "24hVolume",
  },
  {
    Title: "Recommended: ",
    Icon: <ExclamationCircleOutlined />,
    Value: "recommended",
  },
];
```



```
const Exchanges = () => {
  const [curId, setCurId] = useState("Qwsogvtv82FCd");
  const [category, setCategory] = useState("Bitcoin");
  const { data } = useGetCryptosQuery(100);
  const { data: coinData, isLoading } = useGetCoinExchangeQuery(curId);

  useEffect(() => {
    const newData = data?.data?.coins?.filter((coin) => category ===
coin.name);
    setCurId(newData[0]?.uuid);
    // console.log(newData[0].uuid);
  }, [category, data]);

  if (isLoading) {
    return <Loader />;
  }

  const coinExchange = coinData?.data?.exchanges;
  // console.log(coinData);
  // console.log(coinExchange);
  // console.log(data);

  return (
    <>
      <Row>
        <Col span={12}>
          <Title level={2}>Exchanges</Title>
        </Col>
        <Col span={12}>
          <Select
            showSearch
            className="select-news"
            placeholder="Select a Crypto"
            optionFilterProp="children"
            onChange={(value) => setCategory(value)}
            filterOption={(input, option) =>
              option.children.toLowerCase().indexOf(input.toLowerCase())
>= 0
            }
          >
```

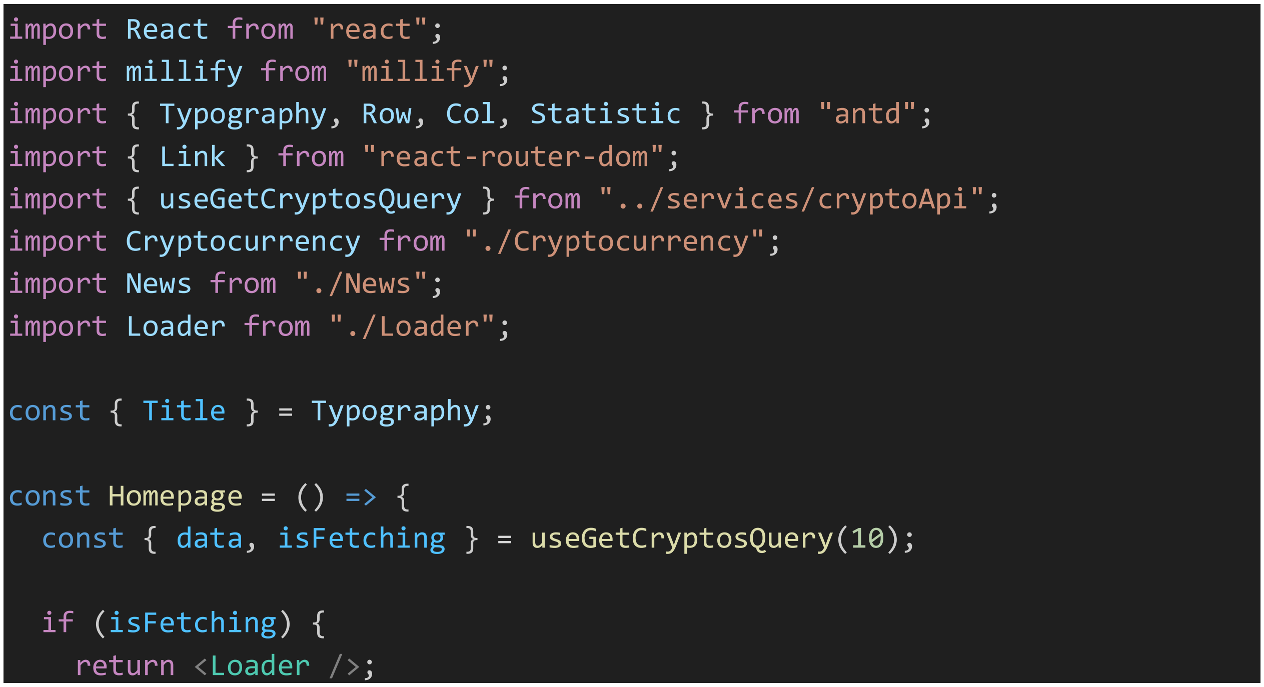


```
      {data?.data?.coins.map((coin) => (
        <Option key={coin.uuid} value={coin.name}>
          {coin.name}
        </Option>
      ))}
    </Select>
  </Col>
</Row>
<Row>
  <Col span={24}>
    <Collapse defaultActiveKey={["1"]}>
      {coinExchange.map((exchange) => (
        <Panel
          key={exchange.uuid}
          showArrow={false}
          header={
            <Row key={exchange.uuid}>
              <Col span={24}>
                <Text>
                  <strong>{exchange.rank}. </strong>
                </Text>
                <Avatar
                  className="exchange-image"
                  src={exchange.iconUrl}
                />
                <Text>
                  <strong>{exchange.name}</strong>
                </Text>
              </Col>
            </Row>
          }
        >
          {genericStats.map((obj) => (
            <Row>
              <Col span={6}>
                <Title level={5}>
                  <Text>{obj.Icon}</Text>
                  <Text> </Text>
                  <Text>
                    <strong>{obj.Title}</strong>
```

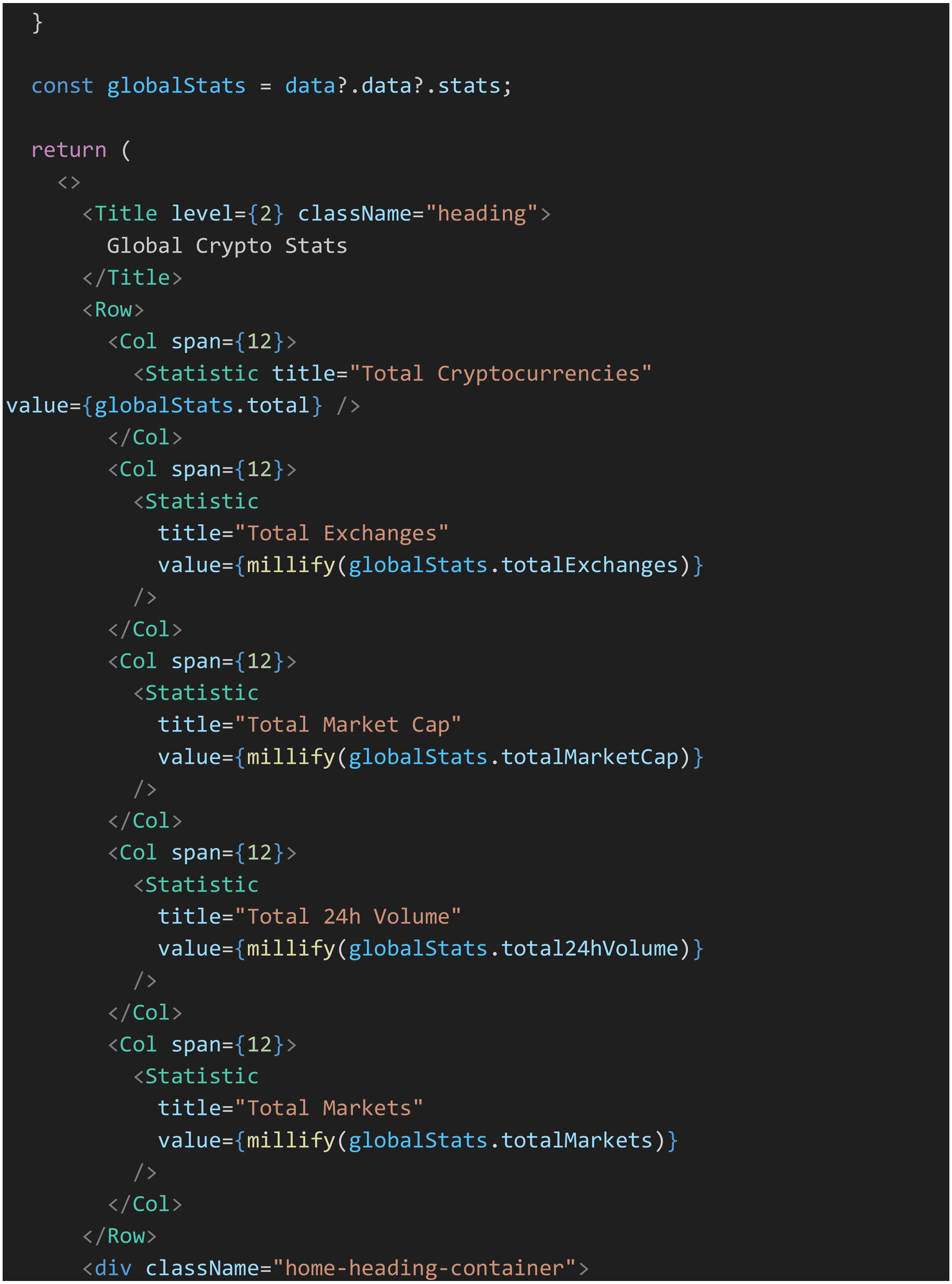


```
                        </Text>
                      </Title>
                    </Col>
                    <Col>
                      {obj.Value !== "recommended"
                        ? exchange[`${obj.Value}`]
                        : "True"}
                    </Col>
                  </Row>
                ))}
              </Panel>
            ))}
          </Collapse>
        </Col>
      </Row>
    </>
  );
};

export default Exchanges;
```

Home page:



```
import React from "react";
import millify from "millify";
import { Typography, Row, Col, Statistic } from "antd";
import { Link } from "react-router-dom";
import { useGetCryptosQuery } from "../services/cryptoApi";
import Cryptocurrency from "./Cryptocurrency";
import News from "./News";
import Loader from "./Loader";

const { Title } = Typography;

const Homepage = () => {
  const { data, isFetching } = useGetCryptosQuery(10);

  if (isFetching) {
    return <Loader />;
```



```
  }

  const globalStats = data?.data?.stats;

  return (
    <>
      <Title level={2} className="heading">
        Global Crypto Stats
      </Title>
      <Row>
        <Col span={12}>
          <Statistic title="Total Cryptocurrencies"
value={globalStats.total} />
        </Col>
        <Col span={12}>
          <Statistic
            title="Total Exchanges"
            value={millify(globalStats.totalExchanges)}
          />
        </Col>
        <Col span={12}>
          <Statistic
            title="Total Market Cap"
            value={millify(globalStats.totalMarketCap)}
          />
        </Col>
        <Col span={12}>
          <Statistic
            title="Total 24h Volume"
            value={millify(globalStats.total24hVolume)}
          />
        </Col>
        <Col span={12}>
          <Statistic
            title="Total Markets"
            value={millify(globalStats.totalMarkets)}
          />
        </Col>
      </Row>
      <div className="home-heading-container">
```

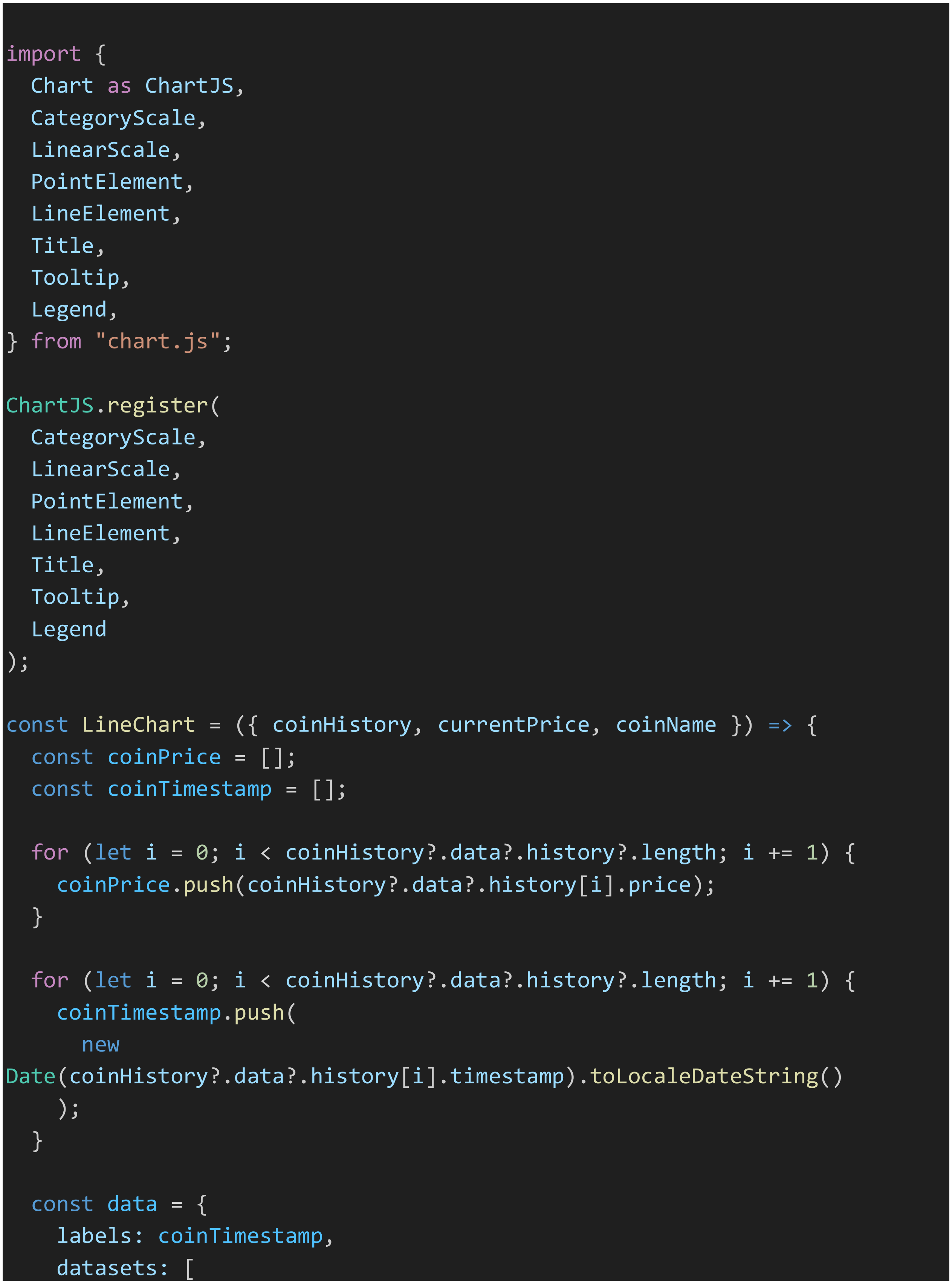
```
        <Title level={2} className="home-title">
          Top 10 Cryptocurrencies in the World
        </Title>
        <Title level={3} className="show-more">
          <Link to="/cryptocurrency">Show More</Link>
        </Title>
      </div>
      <Cryptocurrency simplified />
      <div className="home-heading-container">
        <Title level={2} className="home-title">
          Latest Crypto News
        </Title>
        <Title level={3} className="show-more">
          <Link to="/news">Show More</Link>
        </Title>
      </div>
      <News simplified />
    </>
  );
};

export default Homepage;
```

Index:

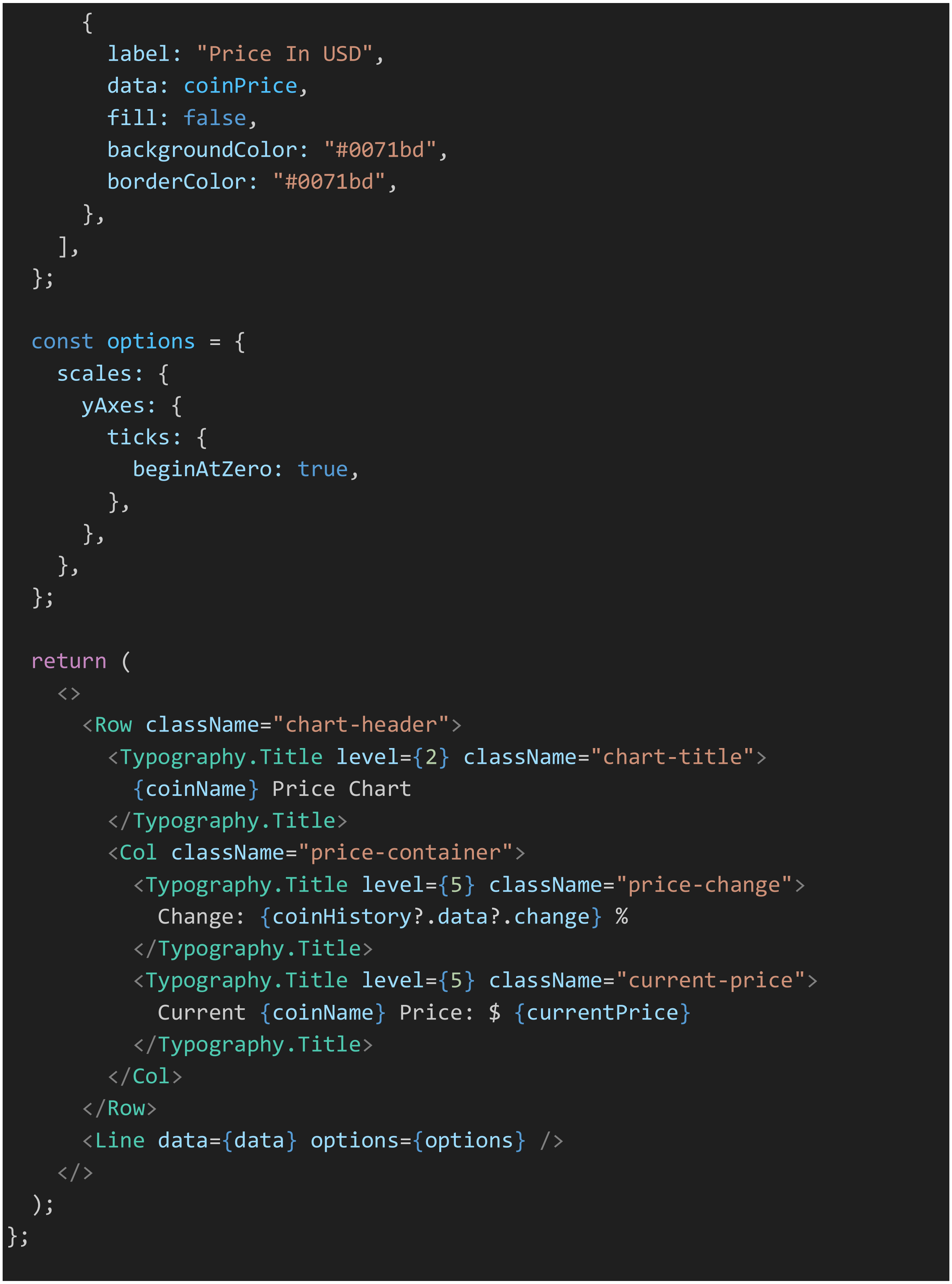
```
export { default as Navbar } from "./Navbar";
export { default as Homepage } from "./Homepage";
export { default as Exchanges } from "./Exchanges";
export { default as Cryptocurrency } from "./Cryptocurrency";
export { default as CryptoDetails } from "./CryptoDetails";
export { default as News } from "./News";
```

LineChart:

```
import React from "react";
import { Line } from "react-chartjs-2";
import { Col, Row, Typography } from "antd";
```



```
import {
  Chart as ChartJS,
  CategoryScale,
  LinearScale,
  PointElement,
  LineElement,
  Title,
  Tooltip,
  Legend,
} from "chart.js";

ChartJS.register(
  CategoryScale,
  LinearScale,
  PointElement,
  LineElement,
  Title,
  Tooltip,
  Legend
);

const LineChart = ({ coinHistory, currentPrice, coinName }) => {
  const coinPrice = [];
  const coinTimestamp = [];

  for (let i = 0; i < coinHistory?.data?.history?.length; i += 1) {
    coinPrice.push(coinHistory?.data?.history[i].price);
  }

  for (let i = 0; i < coinHistory?.data?.history?.length; i += 1) {
    coinTimestamp.push(
      new
Date(coinHistory?.data?.history[i].timestamp).toLocaleDateString()
    );
  }

  const data = {
    labels: coinTimestamp,
    datasets: [
```



```
      {
        label: "Price In USD",
        data: coinPrice,
        fill: false,
        backgroundColor: "#0071bd",
        borderColor: "#0071bd",
      },
    ],
  };

  const options = {
    scales: {
      yAxes: {
        ticks: {
          beginAtZero: true,
        },
      },
    },
  };

  return (
    <>
      <Row className="chart-header">
        <Typography.Title level={2} className="chart-title">
          {coinName} Price Chart
        </Typography.Title>
        <Col className="price-container">
          <Typography.Title level={5} className="price-change">
            Change: {coinHistory?.data?.change} %
          </Typography.Title>
          <Typography.Title level={5} className="current-price">
            Current {coinName} Price: $ {currentPrice}
          </Typography.Title>
        </Col>
      </Row>
      <Line data={data} options={options} />
    </>
  );
};
```

```
export default LineChart;
```

Loader:

```
import React from "react";
import { Spin } from "antd";

const Loader = () => {
  return (
    <div className="loader">
      <Spin />
    </div>
  );
};

export default Loader;
```

Services of js:

cryptoApi:

```
import { createApi, fetchBaseQuery } from "@reduxjs/toolkit/query/react";

const cryptoApiHeaders = {
  "X-RapidAPI-Host": "coinranking1.p.rapidapi.com",
  "X-RapidAPI-Key": "a0482efaddmsh96008daf6627c99p1bfb1bjsn897fc89565d1",
};

const baseUrl = "https://coinranking1.p.rapidapi.com";

const createRequest = (url) => ({ url, headers: cryptoApiHeaders });

export const cryptoApi = createApi({
  reducerPath: "cryptoApi",
  baseQuery: fetchBaseQuery({ baseUrl }),
  endpoints: (builder) => ({
    getCryptos: builder.query({
      query: (count) => createRequest(`/coins?limit=${count}`),
    }),
    getCryptoDetails: builder.query({
      query: (coinId) => createRequest(`/coin/${coinId}`),
    }),
    getCryptoHistory: builder.query({
      query: ({ coinId, timePeriod }) =>
        createRequest(`/coin/${coinId}/history?timperiod=${timePeriod}`),
    }),
    getCoinExchange: builder.query({
      query: (curId) => createRequest(`/coin/${curId}/exchanges`),
    }),
  }),
});

export const {
  useGetCryptosQuery,
  useGetCryptoDetailsQuery,
  useGetCryptoHistoryQuery,
  useGetCoinExchangeQuery,
} = cryptoApi;
```

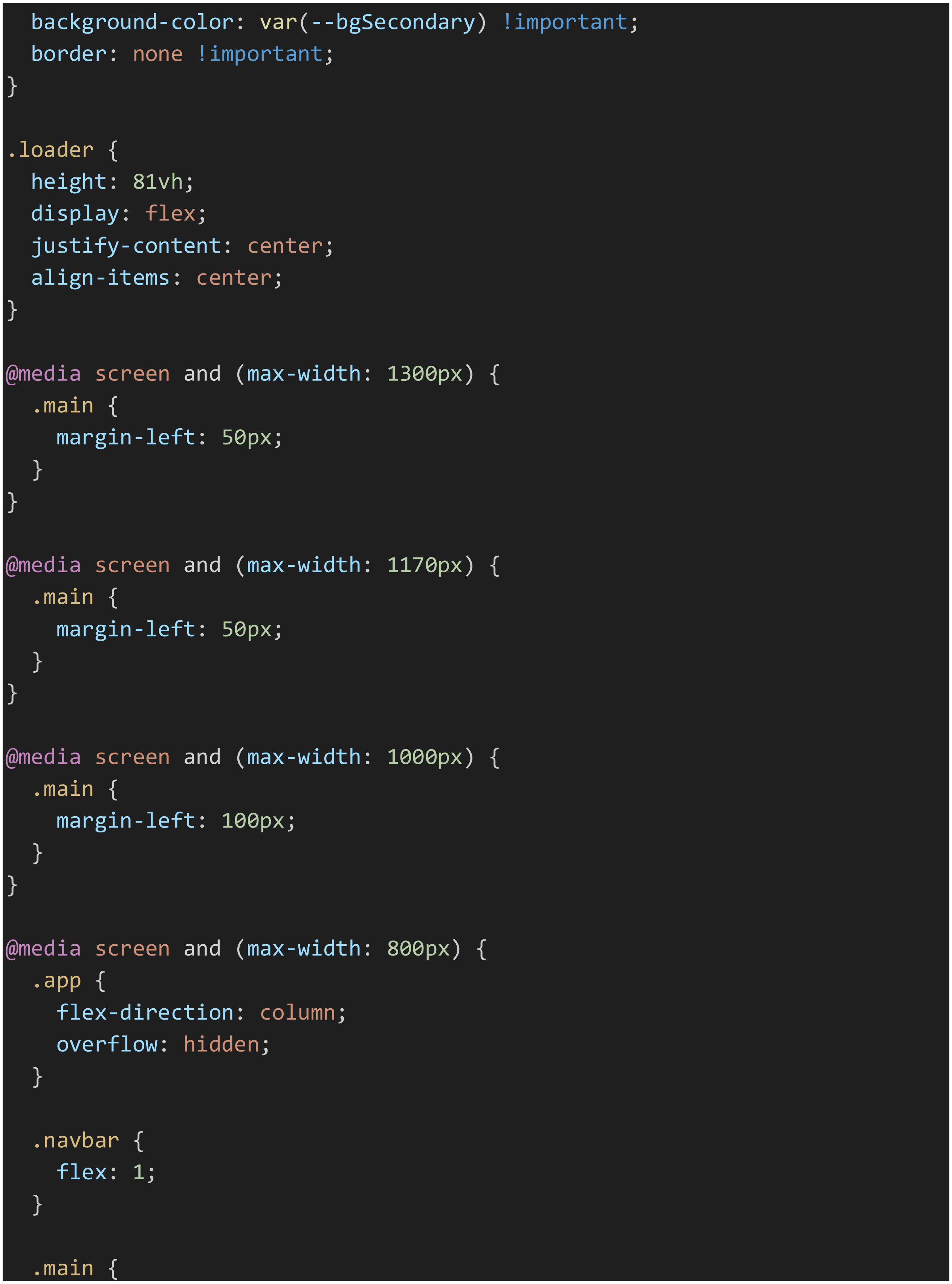
CryptoNewsApi:

```
import { createApi, fetchBaseQuery } from "@reduxjs/toolkit/query/react";

const cryptoNewsApiHeaders = {
  "X-BingApis-SDK": "true",
  "X-RapidAPI-Host": "bing-news-search1.p.rapidapi.com",
  "X-RapidAPI-Key": "a0482efaddmsh96008daf6627c99p1bfb1bjsn897fc89565d1",
};

const baseUrl = "https://bing-news-search1.p.rapidapi.com";

const createRequest = (url) => ({ url, headers: cryptoNewsApiHeaders });

export const cryptoNewsApi = createApi({
  reducerPath: "cryptoNewsApi",
  baseQuery: fetchBaseQuery({ baseUrl }),
  endpoints: (builder) => ({
    getCryptoNews: builder.query({
      query: ({ newsCategory, count }) =>
        createRequest(
          `/news/search?q=${newsCategory}&safeSearch=Off&textFormat=Raw&freshness=Day&count=${count}`
        ),
    }),
  }),
});

export const { useGetCryptoNewsQuery } = cryptoNewsApi;
```

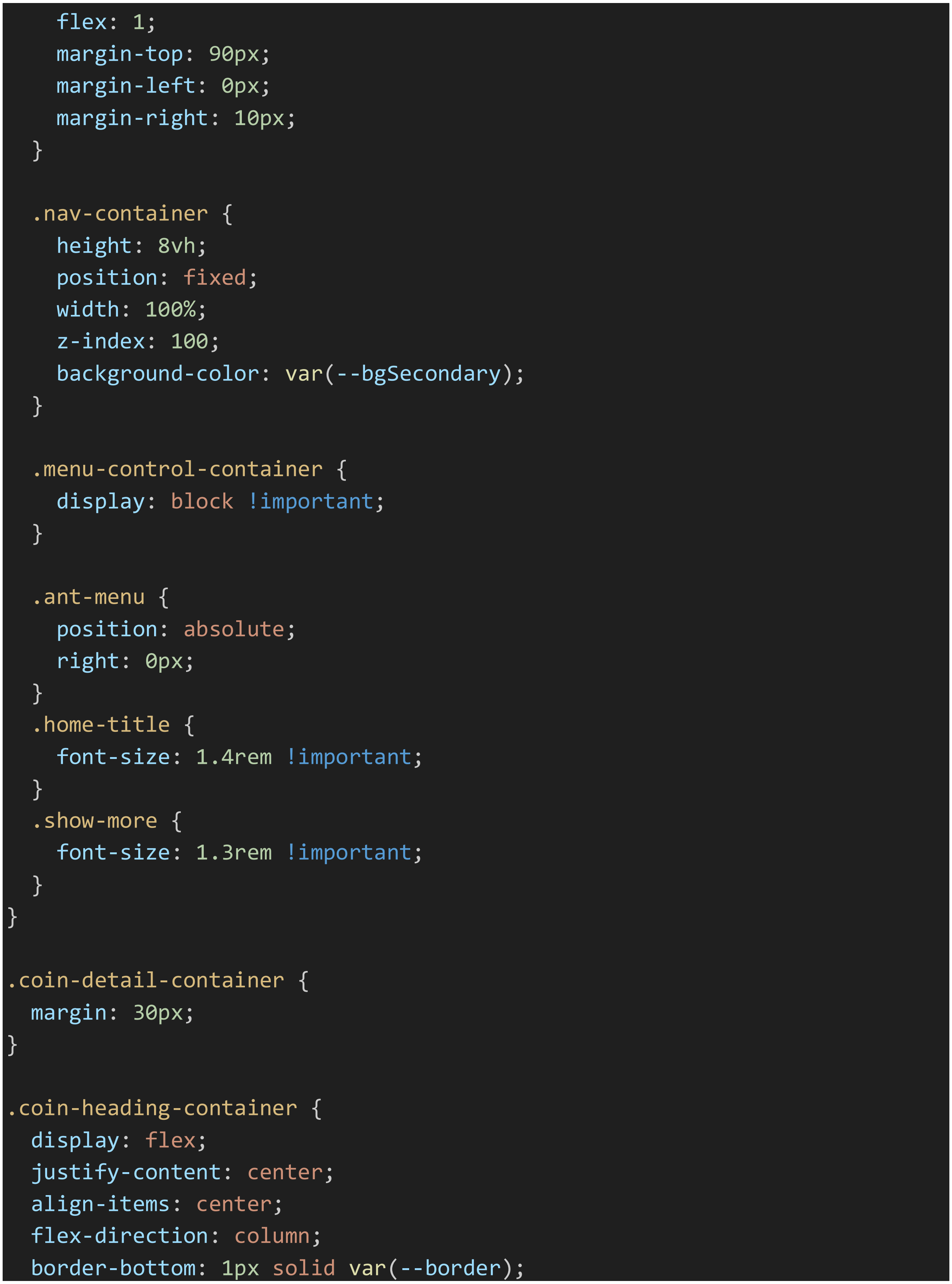
App.cs:

```
:root {
  --text-primary: #000;
  --bgPrimary: #fff;
  --bgSecondary: #f9f9f9;
  --pink: #0071bd;
  --light-blue: #e6f7ff;
  --border: #d9d9d9;
}

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

a:hover {
  color: var(--pink) !important;
}

.app {
  display: flex;
  overflow: hidden;
}
```

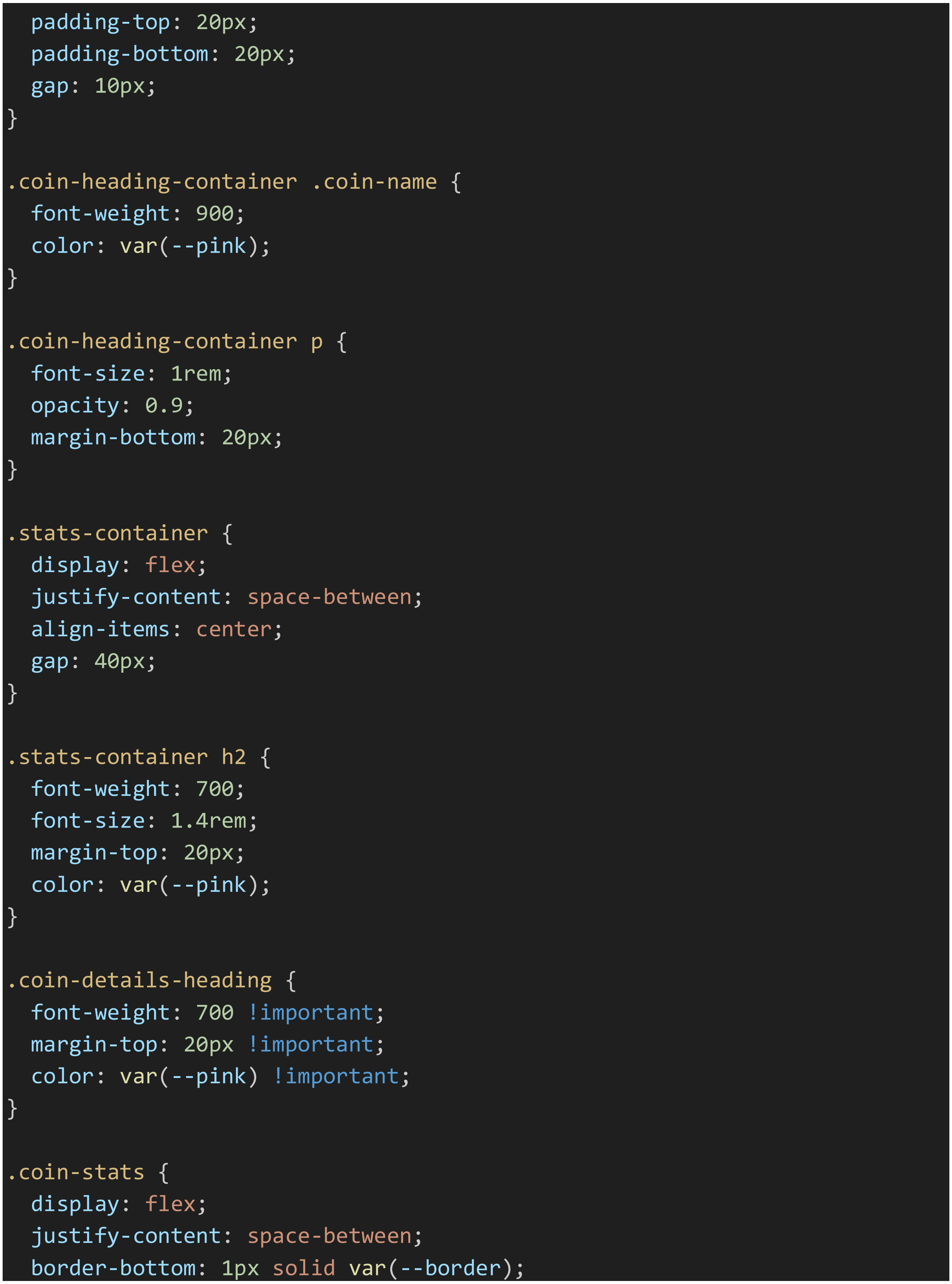
```
.navbar {
  flex: 0.2;
  background-color: rgb(0, 21, 41);
}

.main {
  flex: 0.8;
  width: 100%;
}
.routes {
  padding: 20px;
}
.nav-container {
  position: fixed;
  left: 0;
  margin: 10px;
  height: 100vh;
  margin: 0px;
  background-color: rgb(0, 21, 41);
}

.logo-container {
  background-color: #001529;
  display: flex;
  padding: 20px;
  align-items: center;
  width: 100%;
}
.logo {
  margin: 0 0 0 15px;
}
.logo a {
  color: white;
}
.menu-control-container {
  display: none !important;
  position: absolute !important;
  right: 10px !important;
  top: 25px !important;
  font-size: 1.2rem !important;
```



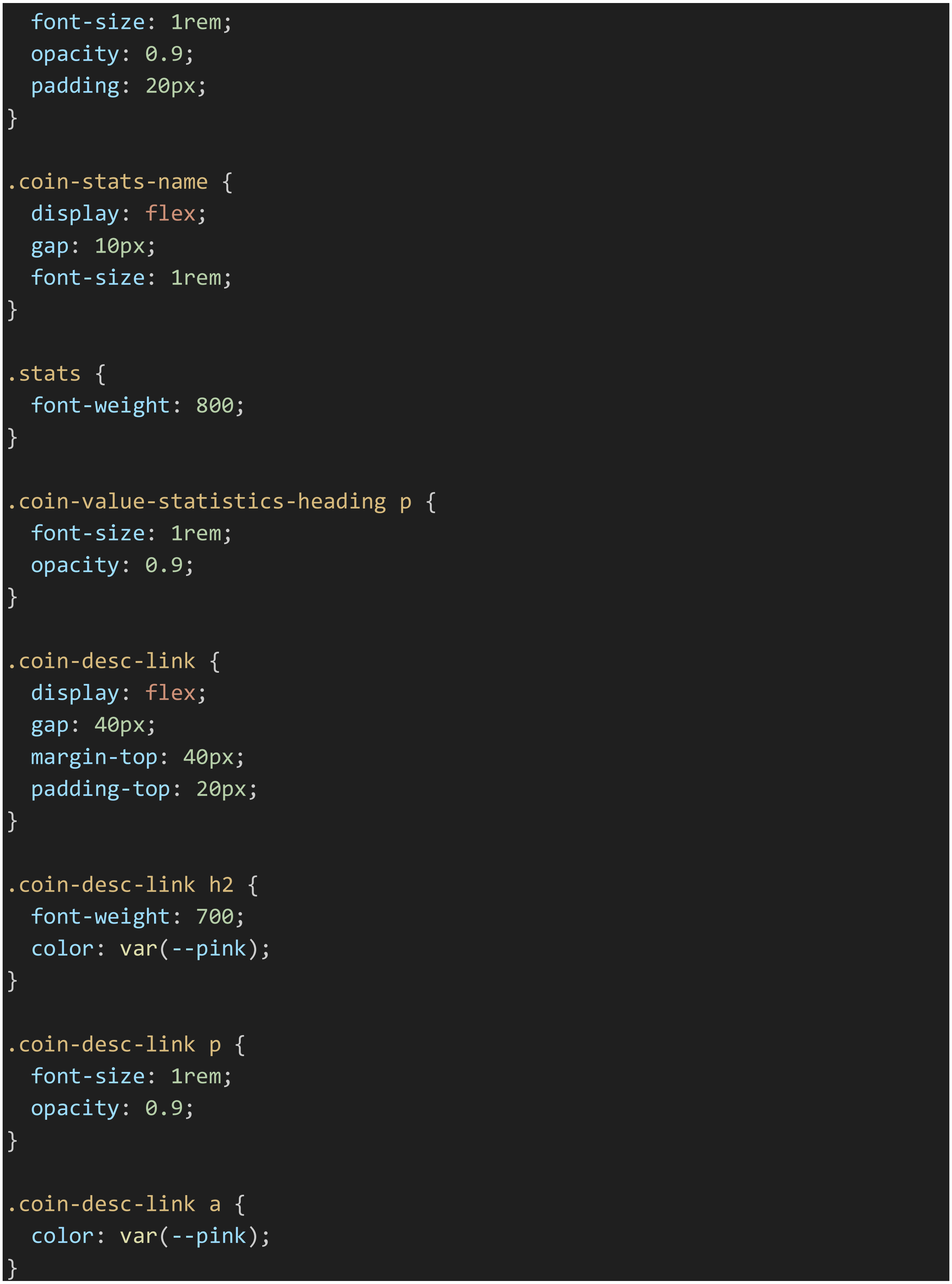
```css
  background-color: var(--bgSecondary) !important;
  border: none !important;
}

.loader {
  height: 81vh;
  display: flex;
  justify-content: center;
  align-items: center;
}

@media screen and (max-width: 1300px) {
  .main {
    margin-left: 50px;
  }
}

@media screen and (max-width: 1170px) {
  .main {
    margin-left: 50px;
  }
}

@media screen and (max-width: 1000px) {
  .main {
    margin-left: 100px;
  }
}

@media screen and (max-width: 800px) {
  .app {
    flex-direction: column;
    overflow: hidden;
  }

  .navbar {
    flex: 1;
  }

  .main {
```



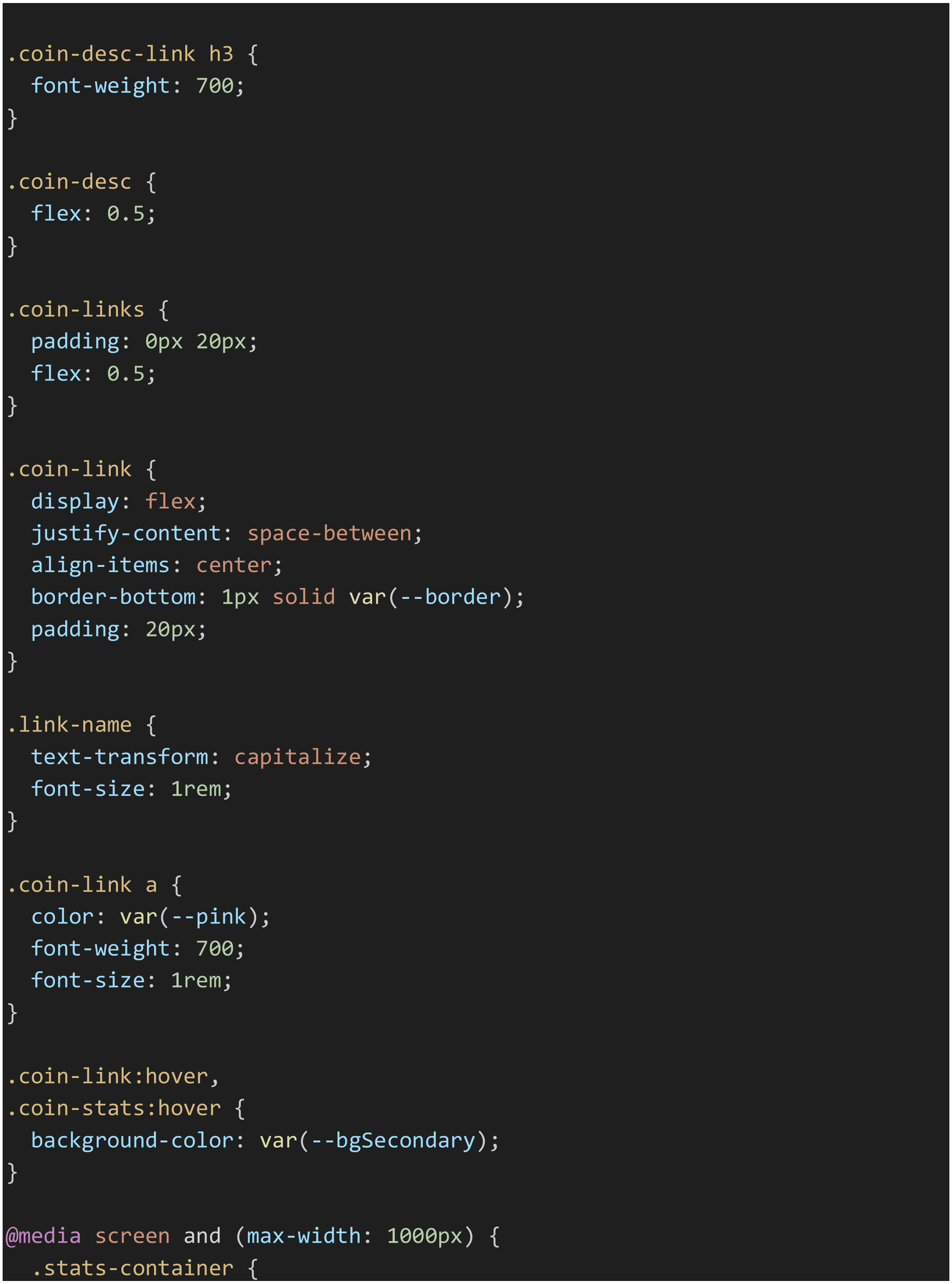
```css
    flex: 1;
    margin-top: 90px;
    margin-left: 0px;
    margin-right: 10px;
  }

  .nav-container {
    height: 8vh;
    position: fixed;
    width: 100%;
    z-index: 100;
    background-color: var(--bgSecondary);
  }

  .menu-control-container {
    display: block !important;
  }

  .ant-menu {
    position: absolute;
    right: 0px;
  }
  .home-title {
    font-size: 1.4rem !important;
  }
  .show-more {
    font-size: 1.3rem !important;
  }
}

.coin-detail-container {
  margin: 30px;
}

.coin-heading-container {
  display: flex;
  justify-content: center;
  align-items: center;
  flex-direction: column;
  border-bottom: 1px solid var(--border);
```



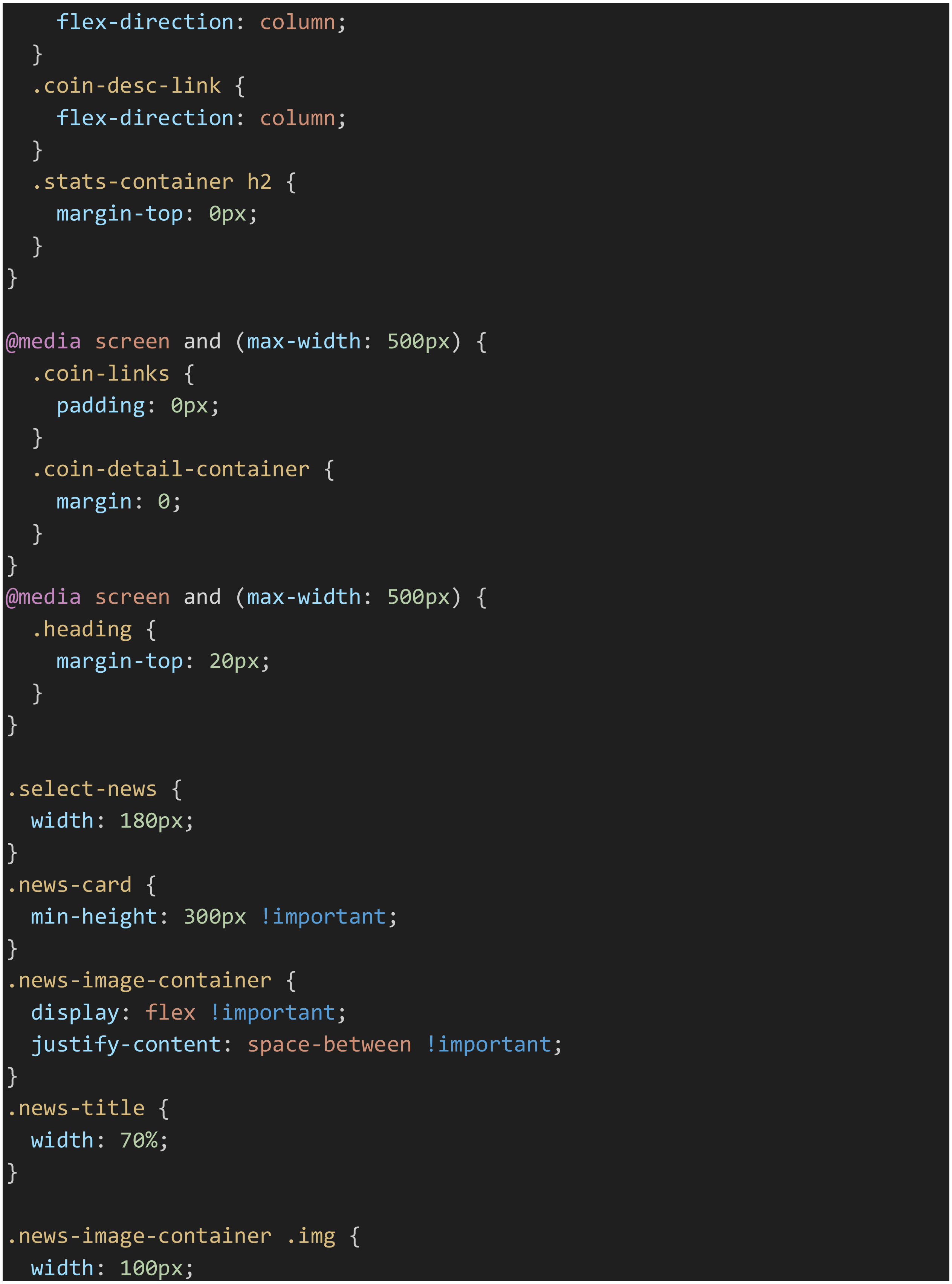
```
  padding-top: 20px;
  padding-bottom: 20px;
  gap: 10px;
}

.coin-heading-container .coin-name {
  font-weight: 900;
  color: var(--pink);
}

.coin-heading-container p {
  font-size: 1rem;
  opacity: 0.9;
  margin-bottom: 20px;
}

.stats-container {
  display: flex;
  justify-content: space-between;
  align-items: center;
  gap: 40px;
}

.stats-container h2 {
  font-weight: 700;
  font-size: 1.4rem;
  margin-top: 20px;
  color: var(--pink);
}

.coin-details-heading {
  font-weight: 700 !important;
  margin-top: 20px !important;
  color: var(--pink) !important;
}

.coin-stats {
  display: flex;
  justify-content: space-between;
  border-bottom: 1px solid var(--border);
```



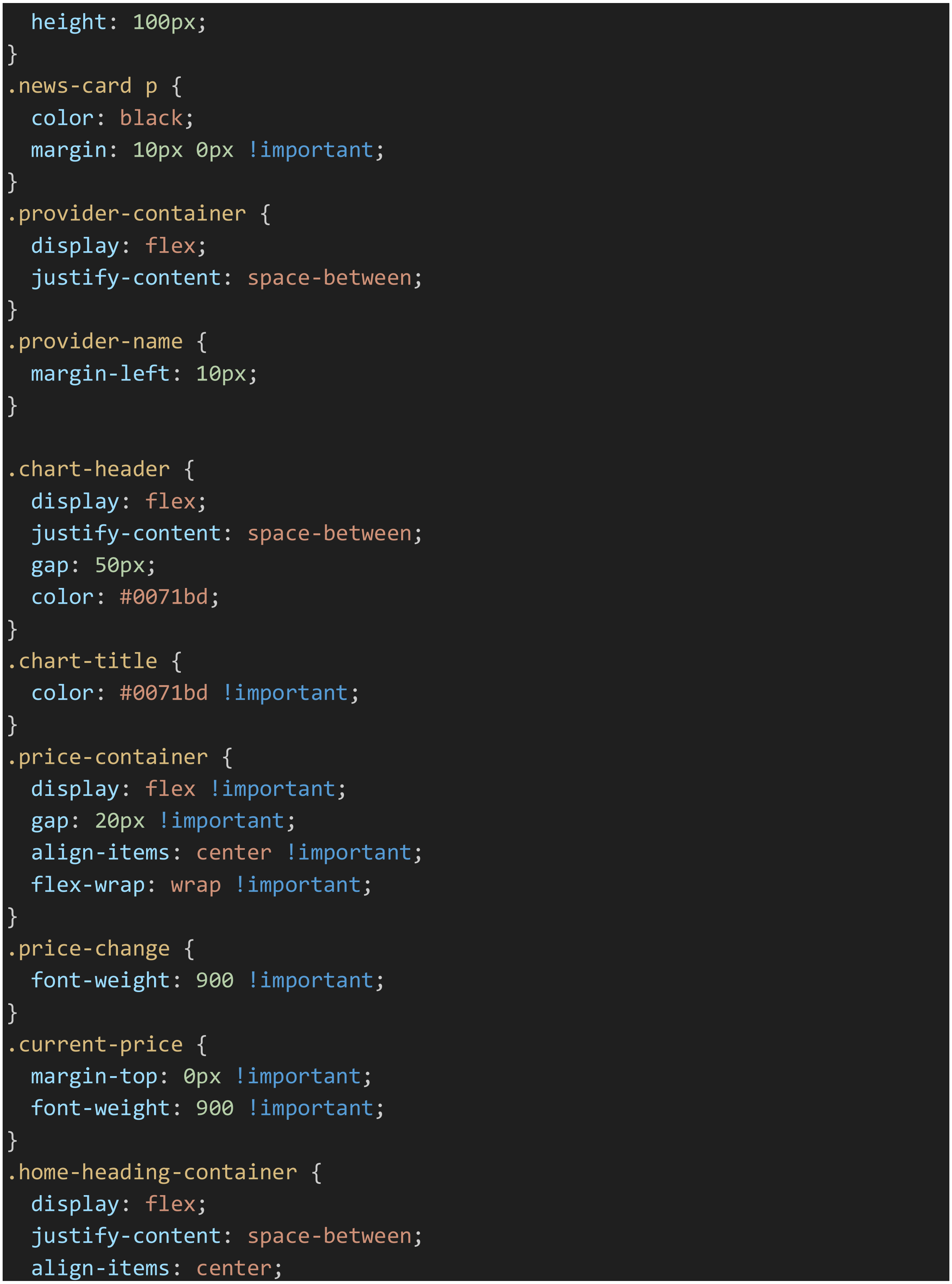
```css
  font-size: 1rem;
  opacity: 0.9;
  padding: 20px;
}

.coin-stats-name {
  display: flex;
  gap: 10px;
  font-size: 1rem;
}

.stats {
  font-weight: 800;
}

.coin-value-statistics-heading p {
  font-size: 1rem;
  opacity: 0.9;
}

.coin-desc-link {
  display: flex;
  gap: 40px;
  margin-top: 40px;
  padding-top: 20px;
}

.coin-desc-link h2 {
  font-weight: 700;
  color: var(--pink);
}

.coin-desc-link p {
  font-size: 1rem;
  opacity: 0.9;
}

.coin-desc-link a {
  color: var(--pink);
}
```



```css
.coin-desc-link h3 {
  font-weight: 700;
}

.coin-desc {
  flex: 0.5;
}

.coin-links {
  padding: 0px 20px;
  flex: 0.5;
}

.coin-link {
  display: flex;
  justify-content: space-between;
  align-items: center;
  border-bottom: 1px solid var(--border);
  padding: 20px;
}

.link-name {
  text-transform: capitalize;
  font-size: 1rem;
}

.coin-link a {
  color: var(--pink);
  font-weight: 700;
  font-size: 1rem;
}

.coin-link:hover,
.coin-stats:hover {
  background-color: var(--bgSecondary);
}

@media screen and (max-width: 1000px) {
  .stats-container {
```



```
    flex-direction: column;
  }
  .coin-desc-link {
    flex-direction: column;
  }
  .stats-container h2 {
    margin-top: 0px;
  }
}

@media screen and (max-width: 500px) {
  .coin-links {
    padding: 0px;
  }
  .coin-detail-container {
    margin: 0;
  }
}
@media screen and (max-width: 500px) {
  .heading {
    margin-top: 20px;
  }
}

.select-news {
  width: 180px;
}
.news-card {
  min-height: 300px !important;
}
.news-image-container {
  display: flex !important;
  justify-content: space-between !important;
}
.news-title {
  width: 70%;
}

.news-image-container .img {
  width: 100px;
```

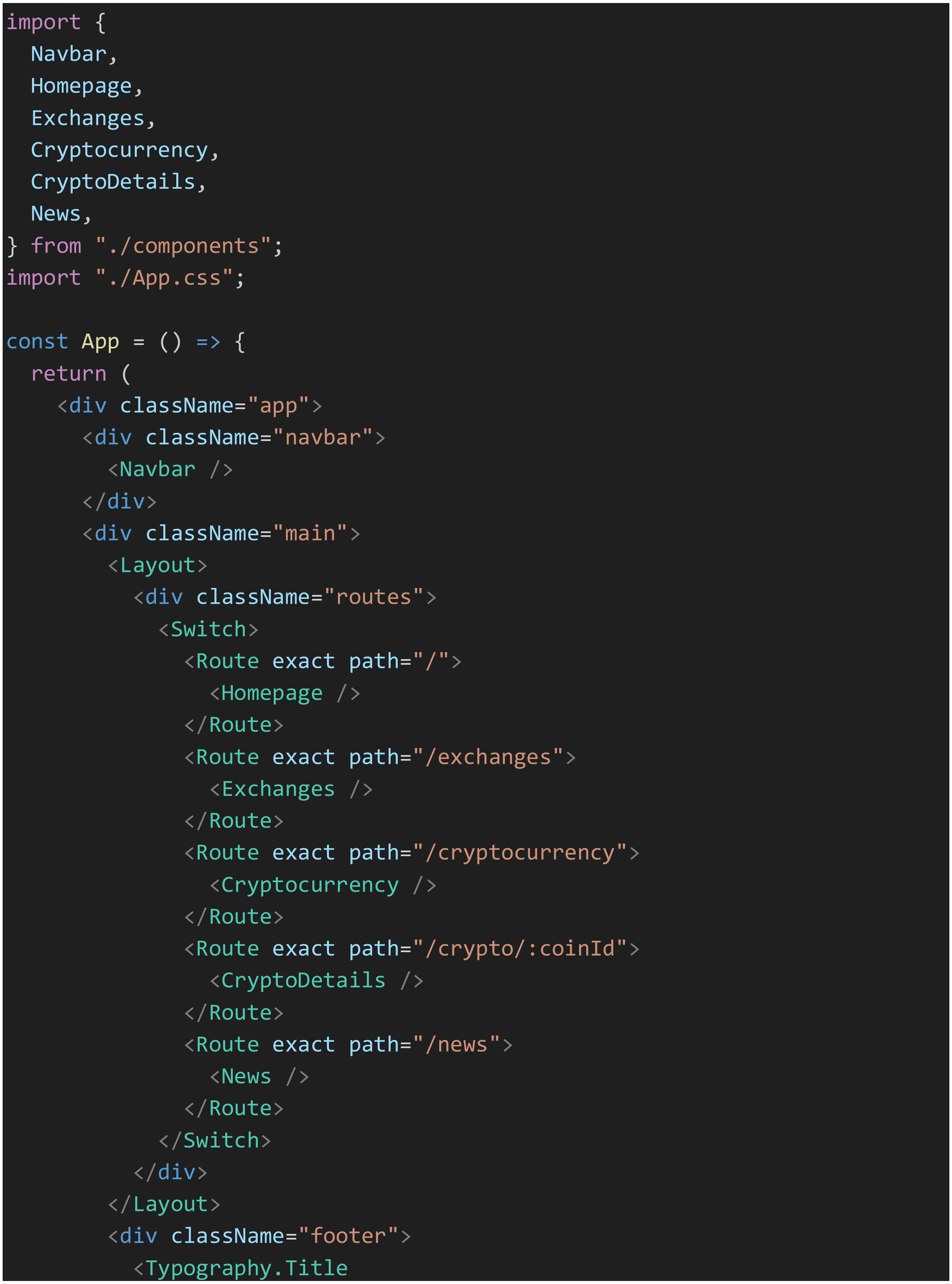


```
  height: 100px;
}
.news-card p {
  color: black;
  margin: 10px 0px !important;
}
.provider-container {
  display: flex;
  justify-content: space-between;
}
.provider-name {
  margin-left: 10px;
}

.chart-header {
  display: flex;
  justify-content: space-between;
  gap: 50px;
  color: #0071bd;
}
.chart-title {
  color: #0071bd !important;
}
.price-container {
  display: flex !important;
  gap: 20px !important;
  align-items: center !important;
  flex-wrap: wrap !important;
}
.price-change {
  font-weight: 900 !important;
}
.current-price {
  margin-top: 0px !important;
  font-weight: 900 !important;
}
.home-heading-container {
  display: flex;
  justify-content: space-between;
  align-items: center;
```

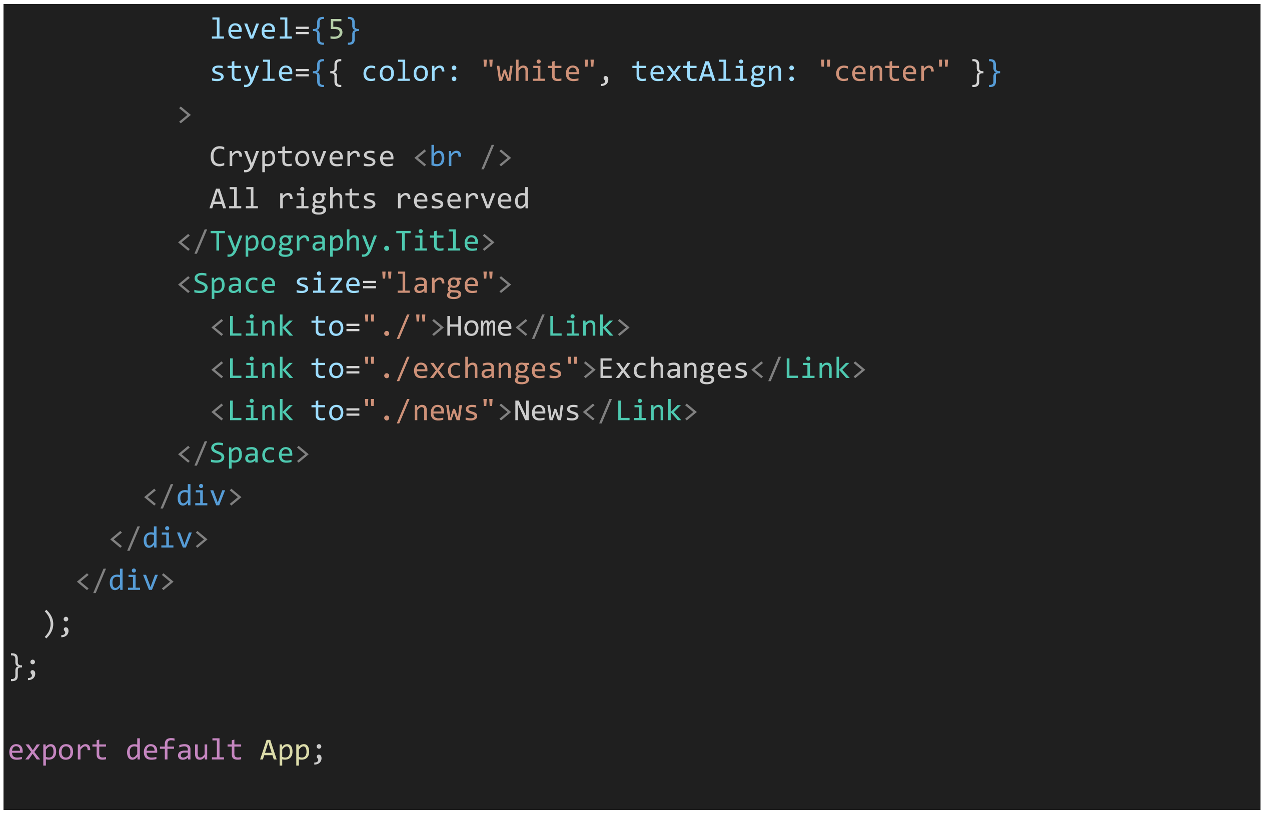
```css
  margin-top: 40px;
}
.show-more {
  margin-top: 0px !important;
}
.exchange-image {
  margin: 0px 10px !important;
}
.search-crypto {
  margin: 20px auto 30px auto;
  width: 250px;
}
.crypto-card-container {
  min-height: 65vh !important;
}
.crypto-card {
  min-width: 250px;
}
.crypto-card .crypto-image {
  width: 35px;
}
.select-timeperiod {
  width: 200px !important;
  margin-top: 20px !important;
}

.footer {
  background-color: #001529;
  display: flex;
  flex-direction: column;
  padding: 20px;
  align-items: center;
}
```

APP.js

```js
import React from "react";
import { Switch, Route, Link } from "react-router-dom";
import { Layout, Typography, Space } from "antd";
```

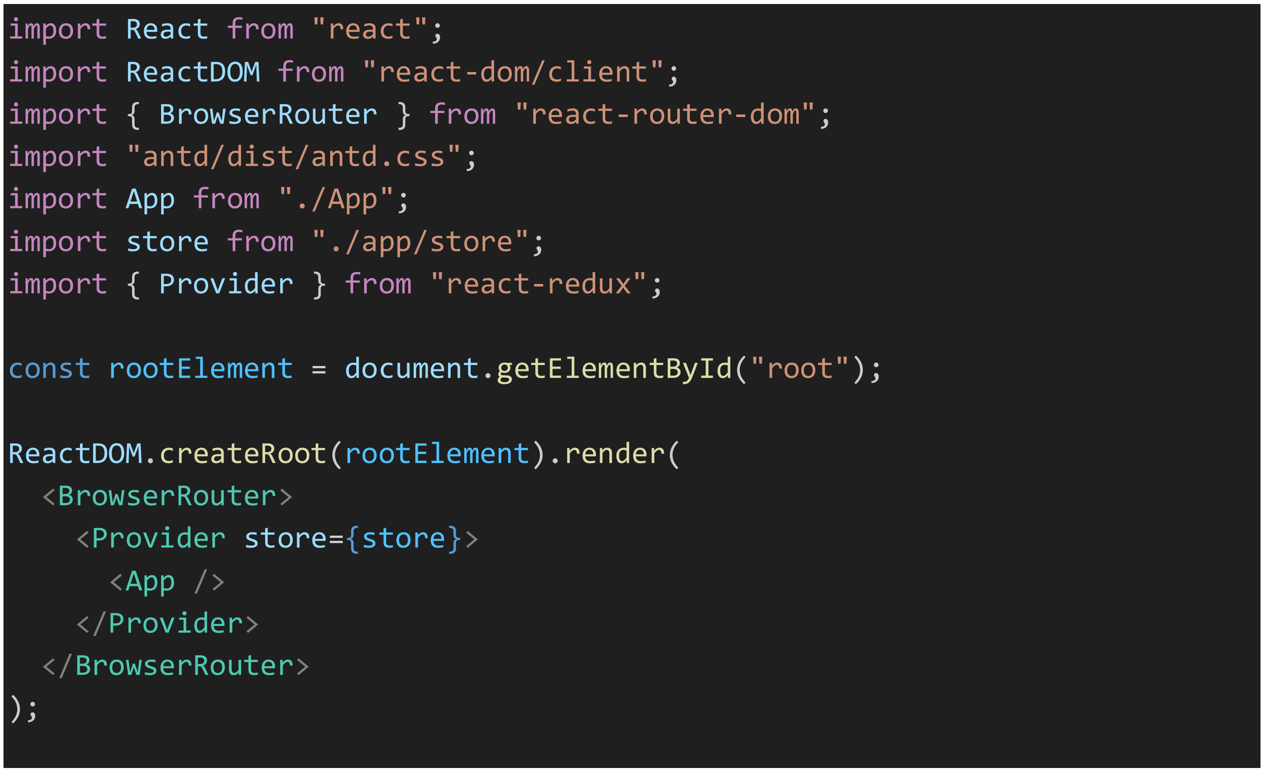


```
import {
  Navbar,
  Homepage,
  Exchanges,
  Cryptocurrency,
  CryptoDetails,
  News,
} from "./components";
import "./App.css";

const App = () => {
  return (
    <div className="app">
      <div className="navbar">
        <Navbar />
      </div>
      <div className="main">
        <Layout>
          <div className="routes">
            <Switch>
              <Route exact path="/">
                <Homepage />
              </Route>
              <Route exact path="/exchanges">
                <Exchanges />
              </Route>
              <Route exact path="/cryptocurrency">
                <Cryptocurrency />
              </Route>
              <Route exact path="/crypto/:coinId">
                <CryptoDetails />
              </Route>
              <Route exact path="/news">
                <News />
              </Route>
            </Switch>
          </div>
        </Layout>
        <div className="footer">
          <Typography.Title
```



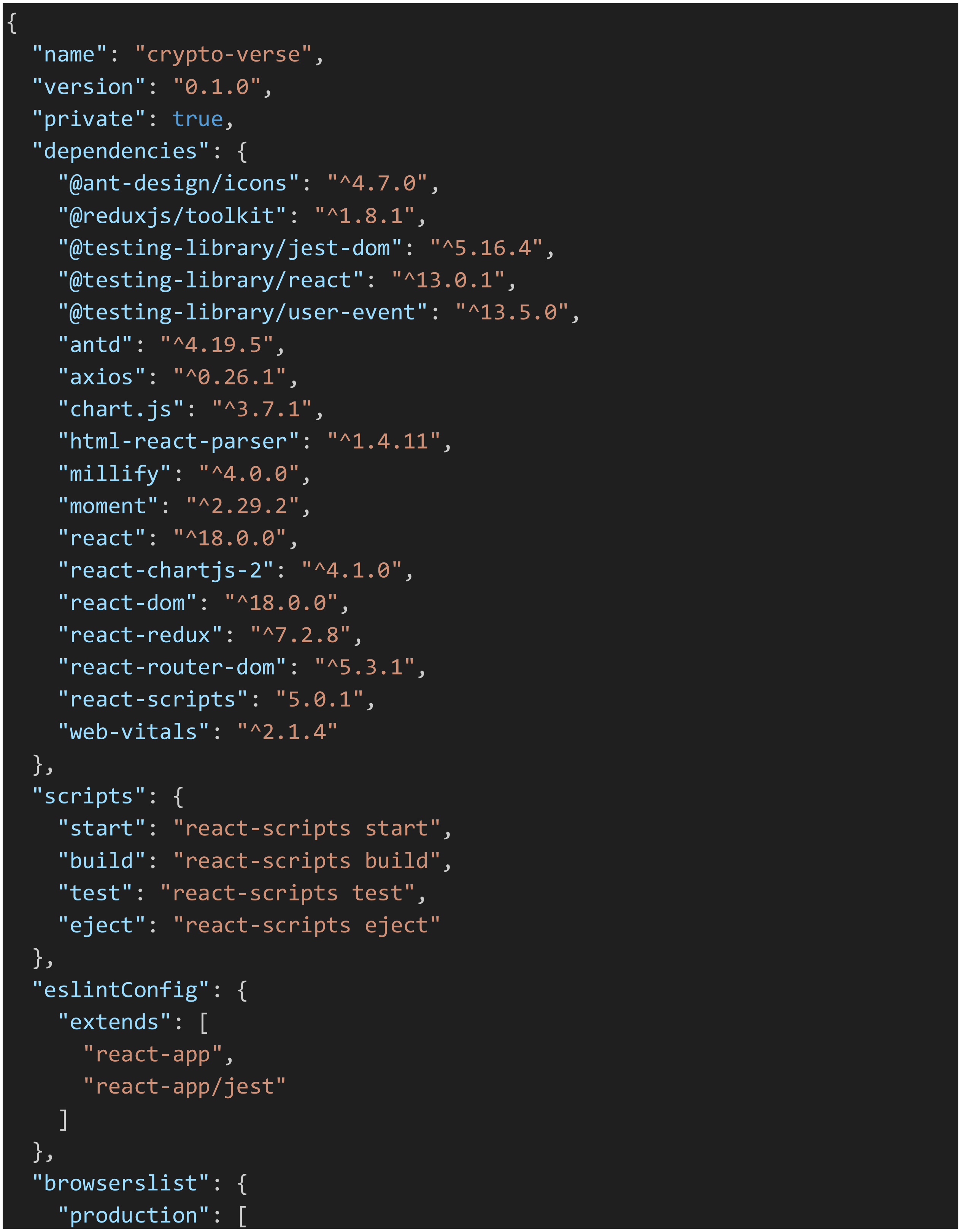
```
            level={5}
            style={{ color: "white", textAlign: "center" }}
          >
            Cryptoverse <br />
            All rights reserved
          </Typography.Title>
          <Space size="large">
            <Link to="./">Home</Link>
            <Link to="./exchanges">Exchanges</Link>
            <Link to="./news">News</Link>
          </Space>
        </div>
      </div>
    </div>
  );
};

export default App;
```

Index.js:



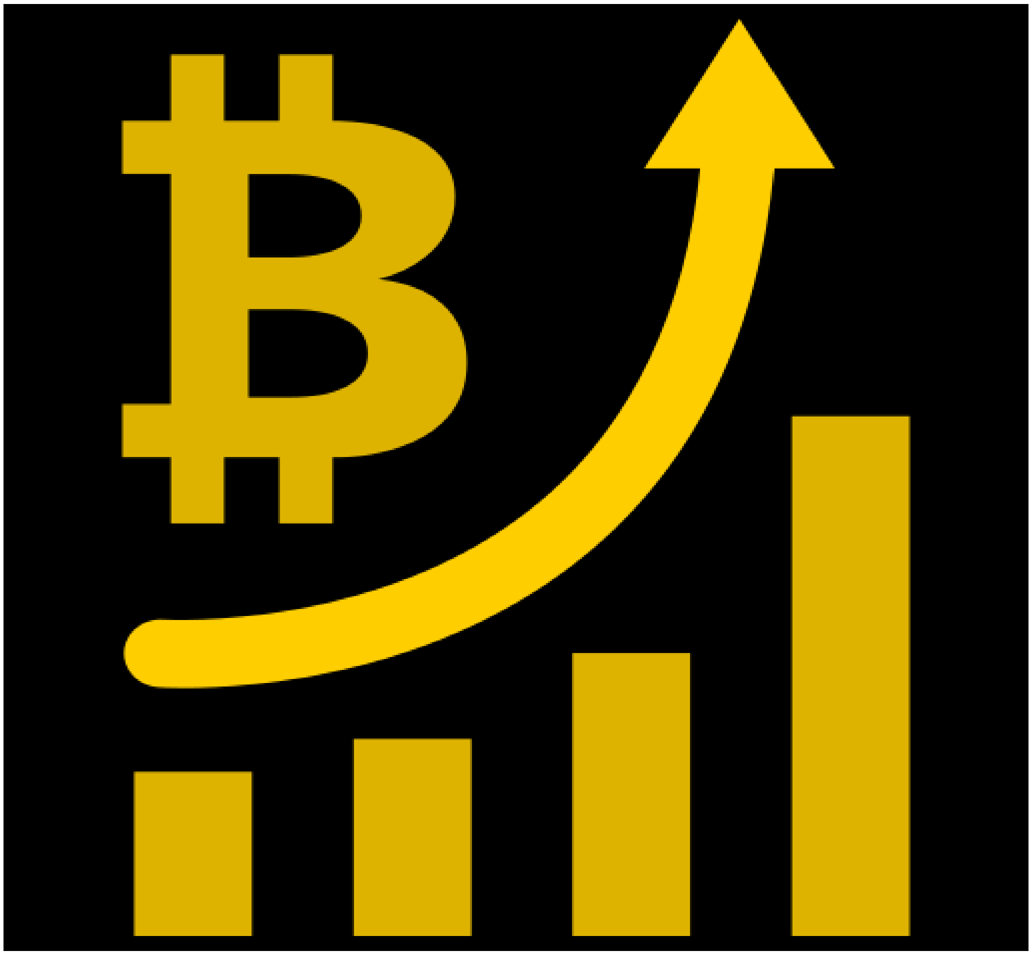
```
import React from "react";
import ReactDOM from "react-dom/client";
import { BrowserRouter } from "react-router-dom";
import "antd/dist/antd.css";
import App from "./App";
import store from "./app/store";
import { Provider } from "react-redux";

const rootElement = document.getElementById("root");

ReactDOM.createRoot(rootElement).render(
  <BrowserRouter>
    <Provider store={store}>
      <App />
    </Provider>
  </BrowserRouter>
);
```

Package.js:

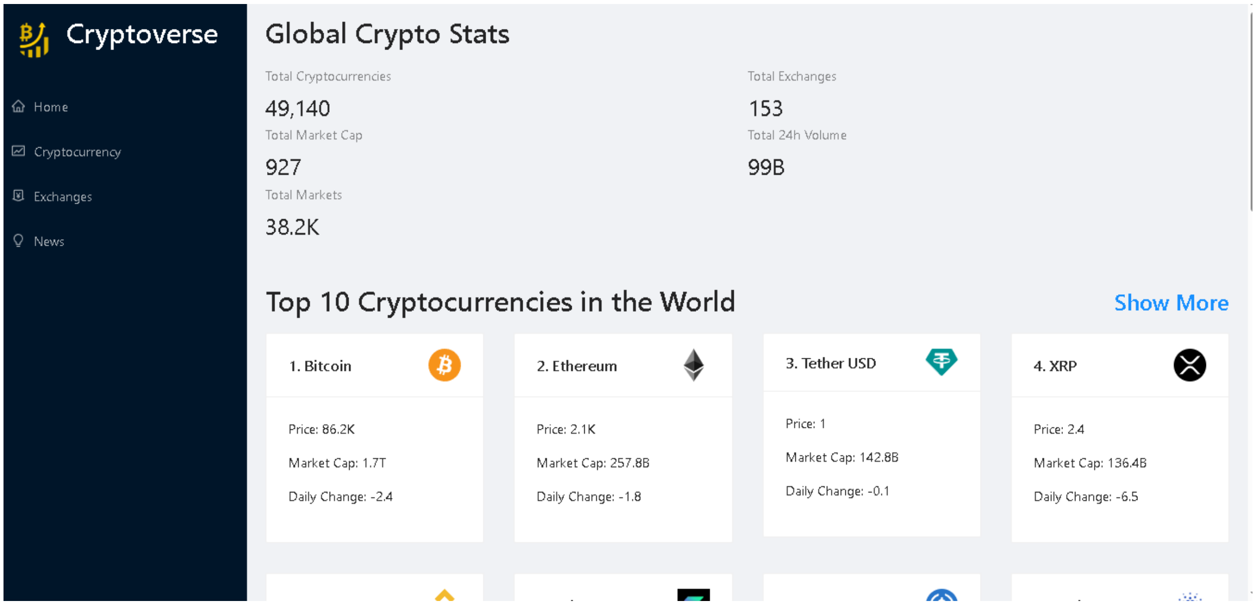


```
{
  "name": "crypto-verse",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@ant-design/icons": "^4.7.0",
    "@reduxjs/toolkit": "^1.8.1",
    "@testing-library/jest-dom": "^5.16.4",
    "@testing-library/react": "^13.0.1",
    "@testing-library/user-event": "^13.5.0",
    "antd": "^4.19.5",
    "axios": "^0.26.1",
    "chart.js": "^3.7.1",
    "html-react-parser": "^1.4.11",
    "millify": "^4.0.0",
    "moment": "^2.29.2",
    "react": "^18.0.0",
    "react-chartjs-2": "^4.1.0",
    "react-dom": "^18.0.0",
    "react-redux": "^7.2.8",
    "react-router-dom": "^5.3.1",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```

Themes:



Screenshots:



Global Crypto Stats

Total Cryptocurrencies
49,140

Total Exchanges
153

Total Market Cap
927

Total 24h Volume
99B

Total Markets
38.2K

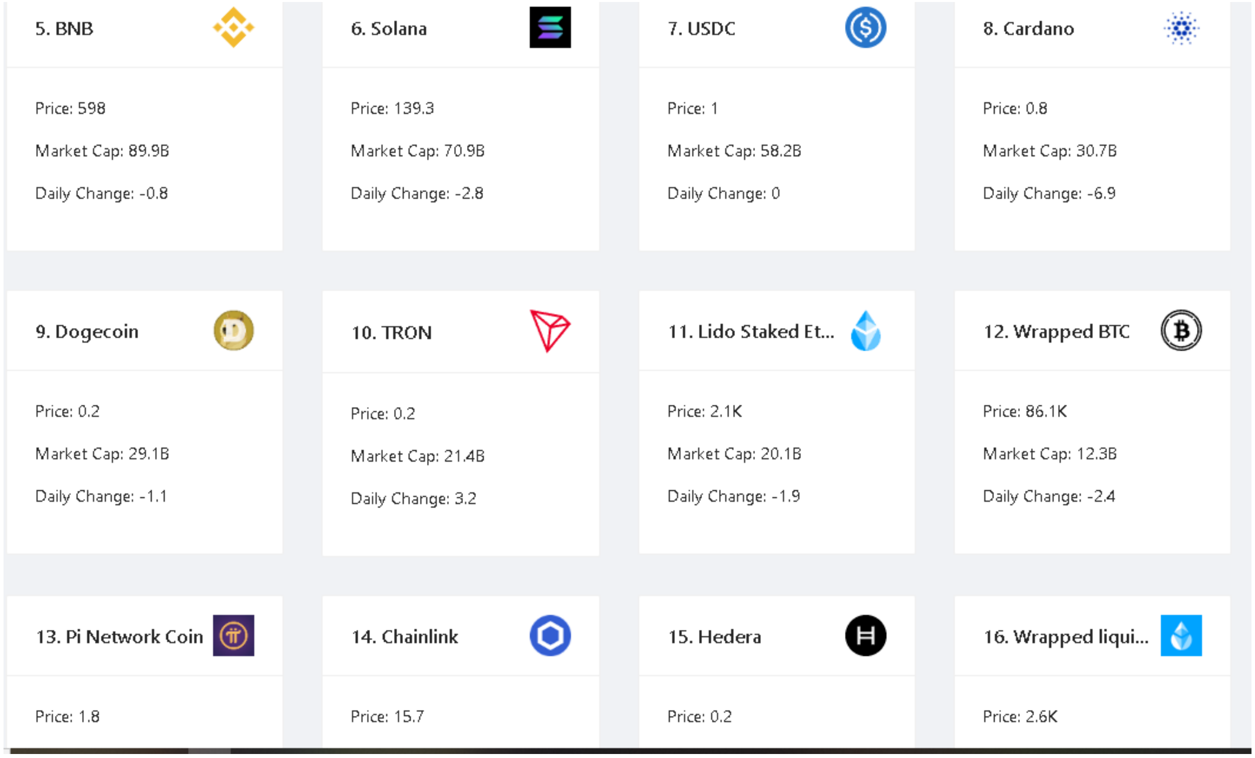
Top 10 Cryptocurrencies in the World

1. Bitcoin
Price: 86.2K
Market Cap: 1.7T
Daily Change: -2.4

2. Ethereum
Price: 2.1K
Market Cap: 257.8B
Daily Change: -1.8

3. Tether USD
Price: 1
Market Cap: 142.8B
Daily Change: -0.1

4. XRP
Price: 2.4
Market Cap: 136.4B
Daily Change: -6.5



5. BNB
Price: 598
Market Cap: 89.9B
Daily Change: -0.8

6. Solana
Price: 139.3
Market Cap: 70.9B
Daily Change: -2.8

7. USDC
Price: 1
Market Cap: 58.2B
Daily Change: 0

8. Cardano
Price: 0.8
Market Cap: 30.7B
Daily Change: -6.9

9. Dogecoin
Price: 0.2
Market Cap: 29.1B
Daily Change: -1.1

10. TRON
Price: 0.2
Market Cap: 21.4B
Daily Change: 3.2

11. Lido Staked Et...
Price: 2.1K
Market Cap: 20.1B
Daily Change: -1.9

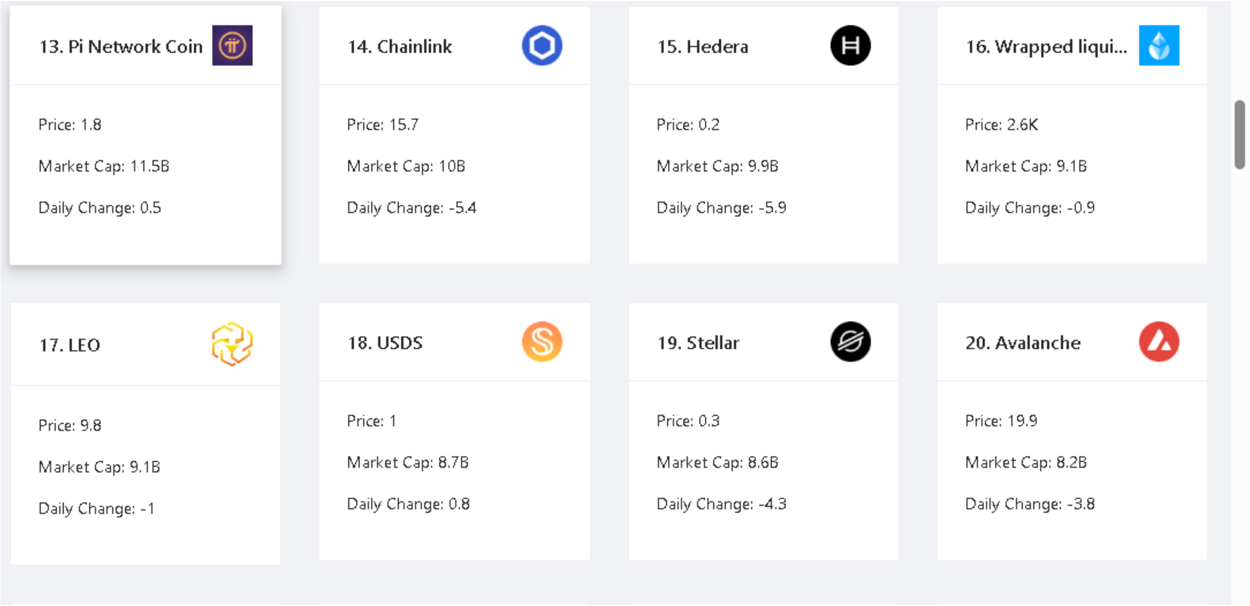
12. Wrapped BTC
Price: 86.1K
Market Cap: 12.3B
Daily Change: -2.4

13. Pi Network Coin
Price: 1.8

14. Chainlink
Price: 15.7

15. Hedera
Price: 0.2

16. Wrapped liqui...
Price: 2.6K



13. Pi Network Coin

Price: 1.8

Market Cap: 11.5B

Daily Change: 0.5

14. Chainlink

Price: 15.7

Market Cap: 10B

Daily Change: -5.4

15. Hedera

Price: 0.2

Market Cap: 9.9B

Daily Change: -5.9

16. Wrapped liqui...

Price: 2.6K

Market Cap: 9.1B

Daily Change: -0.9

17. LEO

Price: 9.8

Market Cap: 9.1B

Daily Change: -1

18. USDS

Price: 1

Market Cap: 8.7B

Daily Change: 0.8

19. Stellar

Price: 0.3

Market Cap: 8.6B

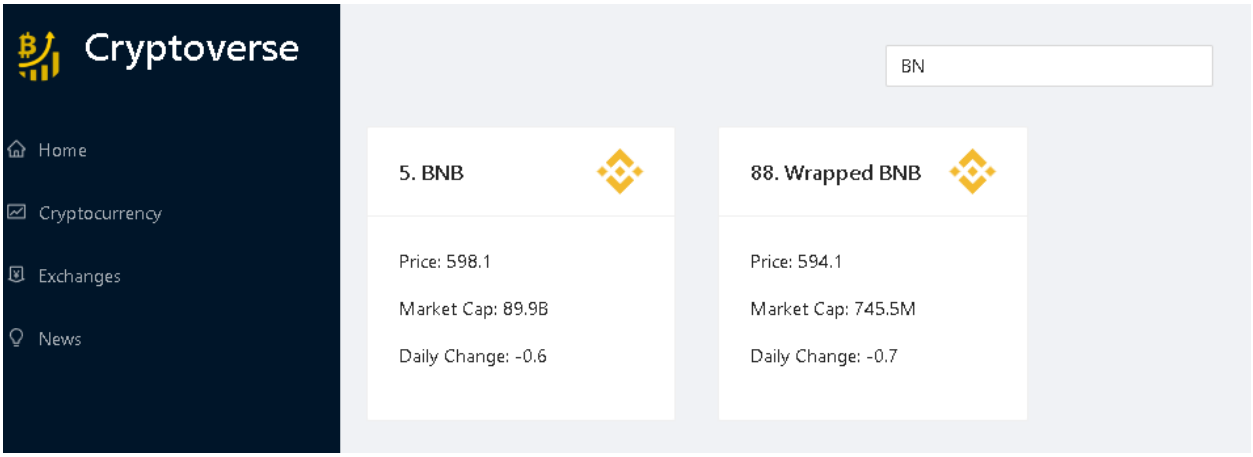
Daily Change: -4.3

20. Avalanche

Price: 19.9

Market Cap: 8.2B

Daily Change: -3.8



BN

5. BNB

Price: 598.1

Market Cap: 89.9B

Daily Change: -0.6

88. Wrapped BNB

Price: 594.1

Market Cap: 745.5M

Daily Change: -0.7

Github repository link:

https://github.com/mithrannjr/crypto-verse

Conclusion:

Cryptoverse is more than just a price-tracking tool—it is a **comprehensive analytics platform** that empowers investors with historical data, interactive insights, and educational resources. By integrating core components such as **historical analysis, risk assessment, and portfolio optimization**, Cryptoverse helps users make **strategic, data-driven investment decisions** in the ever-evolving cryptocurrency market.