

**PROJECT REPORT**

# CRYPTOVERSE

|  |  |
| --- | --- |
| YEAR | : **2024 – 2025** |
| COLLEGE NAME | : **K.C.S KASI NADAR COLLEGE OF ARTS & SCIENCE** |
| CODE | : UNM203 |
| DEPARTMENT | : **COMPUTER SCIENCE** |
| PROGRAM | : **B.Sc Computer Science** |
| SEMESTER | : **VI** |

PROJECT SUBMITTED TO: UNIVERSITY OF MADRAS / NAAN MUDALVAN

Course Name : **Front End Development and Database Administration**

**TEAM LEADER:**

DIWAKAR P

TEAM MEMBERS:

1. KUMARAN J

2. PRAVEEN S

3. BALA YUVARAJ T

4. MUHAMMAD ABDUL RAZAK R

**GUIDED BY: Dr. S.KALAISELVI**

**SPOC NAME: Dr.K. LALITHAKAMESWARI**

**Source Code:**

**Index.html:**

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <link rel="icon" type="image/svg+xml" href="/vite.svg" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Cryptoverse</title>

  </head>

  <body>

    <div id="root"></div>

    <script type="module" src="/src/main.jsx"></script>

  </body>

</html>

**Store,js:**

import { configureStore } from "@reduxjs/toolkit";

import { cryptoApi } from "../services/cryptoApi";

export default configureStore({

  reducer: {

    [cryptoApi.reducerPath]: cryptoApi.reducer,

  },

  middleware: (getDefaultMiddleware) =>

    getDefaultMiddleware().concat(cryptoApi.middleware),

});

**Cryptocurrencies:**

import React, { useEffect, useState } from "react";

import millify from "millify";

import { Link } from "react-router-dom";

import { Card, Row, Col, Input } from "antd";

import { useGetCryptosQuery } from "../services/cryptoApi";

import Loader from "./Loader";

const Cryptocurrencies = ({ simplified }) => {

  const count = simplified ? 10 : 100;

  const { data: cryptosList, isFetching } = useGetCryptosQuery(count);

  const [cryptos, setcryptos] = useState([]);

  const [searchTerm, setsearchTerm] = useState("");

  useEffect(() => {

    const filteredData = cryptosList?.data?.coins.filter((item) =>

      item.name.toLowerCase().includes(searchTerm.toLowerCase())

    );

    setcryptos(filteredData);

  }, [cryptosList, searchTerm]);

  if (isFetching) return <Loader />;

  return (

    <>

      {!simplified && (

        <div className="search-crypto">

          <Input

            placeholder="Search Cryptocurrency"

            onChange={(e) => setsearchTerm(e.target.value)}

          />

        </div>

      )}

      <Row gutter={[32, 32]} className="crypto-card-container">

        {cryptos?.map((currency) => {

          return (

            <Col

              xs={24}

              sm={12}

              lg={6}

              className="crypto-card"

              key={currency.uuid}

            >

              <Link key={currency.uuid} to={`/crypto/${currency.uuid}`}>

                <Card

                  extra={

                    <img className="crypto-image" src={currency.iconUrl} />

                  }

                  title={`${currency.rank}. ${currency.name}`}

                >

                  <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 Cryptocurrencies;

**Cryptodetails.jsx:**

import React, { useEffect, useState } from "react";

import HTMLReactParser from "html-react-parser";

import { useParams } from "react-router-dom";

import millify from "millify";

import { v4 as uuidv4 } from "uuid";

import { Col, Row, Typography, Select } from "antd";

import {

  MoneyCollectOutlined,

  DollarCircleOutlined,

  FundOutlined,

  ExclamationCircleOutlined,

  StopOutlined,

  TrophyOutlined,

  CheckOutlined,

  NumberOutlined,

  ThunderboltOutlined,

} from "@ant-design/icons";

const { Title, Text } = Typography;

const { Option } = Select;

import LineChart from "./LineChart";

import Loader from "../components/Loader";

import {

  useGetCryptoDetailsQuery,

  useGetCryptoHistoryQuery,

} from "../services/cryptoApi";

const CryptoDetails = () => {

  const { coinId } = useParams();

  const [timePeriod, setTimePeriod] = useState("5y");

  const [coinHistory, setCoinHistory] = useState([]);

  const { data: coinHistoryData } = useGetCryptoHistoryQuery({

    coinId,

    timePeriod,

  });

  const { data, isFetching } = useGetCryptoDetailsQuery(coinId);

  useEffect(() => {

    setCoinHistory(coinHistoryData);

    console.log(coinHistoryData);

  }, [coinHistoryData]);

  const cryptoDetails = data?.data?.coin;

  if (isFetching) return <Loader />;

  const time = ["3h", "24h", "7d", "30d", "1y", "3m", "3y", "5y"];

  const volume = data?.data?.coin["24hVolume"];

  const stats = [

    {

      title: "Price to USD",

      value: `$ ${cryptoDetails?.price && millify(cryptoDetails?.price)}`,

      icon: <DollarCircleOutlined />,

    },

    { title: "Rank", value: cryptoDetails?.rank, icon: <NumberOutlined /> },

    {

      title: "24h Volume",

      value: `$ ${volume && millify(volume)}`,

      icon: <ThunderboltOutlined />,

    },

    {

      title: "Market Cap",

      value: `$ ${

        cryptoDetails?.marketCap && millify(cryptoDetails?.marketCap)

      }`,

      icon: <DollarCircleOutlined />,

    },

    {

      title: "All-time-high(daily avg.)",

      value: `$ ${

        cryptoDetails?.allTimeHigh?.price &&

        millify(cryptoDetails?.allTimeHigh?.price)

      }`,

      icon: <TrophyOutlined />,

    },

  ];

  const genericStats = [

    {

      title: "Number Of Markets",

      value: cryptoDetails?.numberOfMarkets,

      icon: <FundOutlined />,

    },

    {

      title: "Number Of Exchanges",

      value: cryptoDetails?.numberOfExchanges,

      icon: <MoneyCollectOutlined />,

    },

    {

      title: "Aprroved Supply",

      value: cryptoDetails?.supply?.confirmed ? (

        <CheckOutlined />

      ) : (

        <StopOutlined />

      ),

      icon: <ExclamationCircleOutlined />,

    },

    {

      title: "Total Supply",

      value: `$ ${

        cryptoDetails?.supply?.total && millify(cryptoDetails?.supply?.total)

      }`,

      icon: <ExclamationCircleOutlined />,

    },

    {

      title: "Circulating Supply",

      value: `$ ${

        cryptoDetails?.supply?.circulating &&

        millify(cryptoDetails?.supply?.circulating)

      }`,

      icon: <ExclamationCircleOutlined />,

    },

  ];

  return (

    <Col className="coin-detail-container">

      <Col className="coin-heading-container">

        <Title level={2} className="coin-name">

          {cryptoDetails?.name}

        </Title>

        <p>

          {cryptoDetails?.name} live price in US dollars. view value statistics,

          market cap and supply.

        </p>

      </Col>

      <Select

        defaultValue="5y"

        className="select-timeperiod"

        placeholder="Select Time Period"

        onChange={(value) => setTimePeriod(value)}

      >

        {time.map((date) => (

          <Option key={date}>{date}</Option>

        ))}

      </Select>

      <LineChart

        currentPrice={millify(cryptoDetails?.price)}

        coinHistory={coinHistory}

        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 }) => {

            return (

              <Col key={uuidv4()} className="coin-stats">

                <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">

              Others statistics

            </Title>

            <p>An overview showing the stats of all cryptocurrencies</p>

          </Col>

          {genericStats.map(({ icon, title, value }) => {

            return (

              <Col key={uuidv4()} className="coin-stats">

                <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">

        <Row className="coin-desc">

          <Title level={3} className="coin-details-heading">

            What is {cryptoDetails?.name} ? <br />

            {HTMLReactParser(cryptoDetails?.description)}

          </Title>

        </Row>

        <Col className="coin-links">

          <Title level={3} className="coin-details-heading">

            {cryptoDetails?.name} Links

          </Title>

          {cryptoDetails?.links.map((link) => {

            return (

              <Row className="coin-link" key={uuidv4()}>

                <Title level={5} className="link-name">

                  {link.type}

                </Title>

                <a href={link.url} target="\_blank" rel="noreferrer">

                  {link.name}

                </a>

              </Row>

            );

          })}

        </Col>

      </Col>

    </Col>

  );

};

export default CryptoDetails;

**Home.js:**

import React from "react";

import milify from "millify";

import { Typography, Row, Col, Statistic } from "antd";

import { Link } from "react-router-dom";

const { Title } = Typography;

import { useGetCryptosQuery } from "../services/cryptoApi";

import Cryptocurrencies from "./Cryptocurrencies";

import Loader from "./Loader";

const Home = () => {

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

console.log("Fetched Data:", data); // Debugging API response

  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 || 0} />

        </Col>

        <Col span={12}>

          <Statistic

            title="Total Exchanges"

            value={milify(globalStats?.totalExchanges || 0)}

          />

        </Col>

        <Col span={12}>

          <Statistic

            title="Total Market Cap"

            value={milify(globalStats?.totalMarketCap || 0)}

          />

        </Col>

        <Col span={12}>

          <Statistic

            title="Total 24h Volume"

            value={milify(globalStats?.total24hVolume || 0)}

          />

        </Col>

        <Col span={12}>

          <Statistic

            title="Total Markets"

            value={milify(globalStats?.totalMarkets || 0)}

          />

        </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="/cryptocurrencies">Show More</Link>

        </Title>

      </div>

      <Cryptocurrencies simplified />

    </>

  );

};

export default Home;

**Index.js:**

export { default as Navbar } from "./Navbar";

export { default as Cryptocurrencies } from "./Cryptocurrencies.jsx";

export { default as CryptoDetails } from "./CryptoDetails.jsx";

export { default as Home } from "./Home.jsx";

**LineChart,jsx:**

**import React from "react";**

**import { Line } from "react-chartjs-2";**

**import { Col, Row, Typography } from "antd";**

**import { Chart as ChartJS } from "chart.js/auto";**

**const { Title } = Typography;**

**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);**

**coinTimestamp.push(**

**new Date(**

**coinHistory?.data?.history[i].timestamp \* 1000**

**).toLocaleDateString()**

**);**

**}**

**const data = {**

**labels: coinTimestamp,**

**datasets: [**

**{**

**label: "Price In USD",**

**data: coinPrice,**

**backgroundColor: "#0071bd",**

**borderColor: "#0071bd",**

**},**

**],**

**};**

**return (**

**<>**

**<Row className="chart-header">**

**<Title level={2} className="chart-title">**

**{coinName} Price Chart**

**</Title>**

**<Col className="price-container">**

**<Title level={5} className="price-change">**

**Change: {coinHistory?.data?.change}%**

**</Title>**

**<Title level={5} className="current-price">**

**Current {coinName} Price: $ {currentPrice}**

**</Title>**

**</Col>**

**</Row>**

**<Line className="chart" data={data} />**

**</>**

**);**

**};**

**export default LineChart;**

**Loader.jsx:**

**import React from "react";**

**import { Spin } from "antd";**

**const Loader = () => {**

**return (**

**<div className="loader">**

**<Spin />**

**</div>**

**);**

**};**

**export default Loader;**

**Navbar.jsx:**

**import React from "react";**

**import { Spin } from "antd";**

**const Loader = () => {**

**return (**

**<div className="loader">**

**<Spin />**

**</div>**

**);**

**};**

**export default Loader;**

**App.jsx:**

**import React from "react";**

**import { Routes, Route, Link } from "react-router-dom";**

**import { Layout } from "antd";**

**import { Navbar, CryptoDetails, Cryptocurrencies, Home } from "./components";**

**import "./App.css";**

**const App = () => {**

**return (**

**<div className="app">**

**<div className="navbar">**

**<Navbar />**

**</div>**

**<div className="main">**

**<Layout>**

**<div className="routes">**

**<Routes>**

**<Route path="/" element={<Home />} />**

**<Route element={<Cryptocurrencies />} path="/cryptocurrencies" />**

**<Route element={<CryptoDetails />} path="/crypto/:coinId" />**

**</Routes>**

**</div>**

**</Layout>**

**<div className="footer">**

**<h1 className="footer-heading">**

**Beyond the Banks: The Rise of Cryptocurrency <br />**

**</h1>**

**</div>**

**</div>**

**</div>**

**);**

**};**

**export default App;**

**Main.jsx:**

**import React from "react";**

**import ReactDOM from "react-dom/client";**

**import App from "./App.jsx";**

**import "./index.css";**

**import { BrowserRouter } from "react-router-dom";**

**import { Provider } from "react-redux";**

**import store from "./app/store.js";**

**ReactDOM.createRoot(document.getElementById("root")).render(**

**<React.StrictMode>**

**<BrowserRouter>**

**<Provider store={store}>**

**<App />**

**</Provider>**

**</BrowserRouter>**

**</React.StrictMode>**

**);**

**Env.local:**

**VITE\_RAPID\_API\_KEY=f6eb15b09bmshc8fdf9749ff17f1p1fcb17jsnfabc6969709a**

**VITE\_RAPID\_API\_HOST=coinranking1.p.rapidapi.com**

**VITE\_BASE\_URL=https://coinranking1.p.rapidapi.com**

**Enlitric:**

**module.exports = {**

**root: true,**

**env: { browser: true, es2020: true },**

**extends: [**

**'eslint:recommended',**

**'plugin:react/recommended',**

**'plugin:react/jsx-runtime',**

**'plugin:react-hooks/recommended',**

**],**

**ignorePatterns: ['dist', '.eslintrc.cjs'],**

**parserOptions: { ecmaVersion: 'latest', sourceType: 'module' },**

**settings: { react: { version: '18.2' } },**

**plugins: ['react-refresh'],**

**rules: {**

**'react-refresh/only-export-components': [**

**'warn',**

**{ allowConstantExport: true },**

**],**

**},**

**}**

**Gitigonre:**

**# Logs**

**logs**

**\*.log**

**npm-debug.log\***

**yarn-debug.log\***

**yarn-error.log\***

**pnpm-debug.log\***

**lerna-debug.log\***

**node\_modules**

**.env**

**dist**

**dist-ssr**

**\*.local**

**# Editor directories and files**

**.vscode/\***

**!.vscode/extensions.json**

**.idea**

**.DS\_Store**

**\*.suo**

**\*.ntvs\***

**\*.njsproj**

**\*.sln**

**\*.sw?**

**Package.json:**

**{**

**"name": "crypto",**

**"private": true,**

**"version": "0.0.0",**

**"type": "module",**

**"scripts": {**

**"dev": "vite",**

**"build": "vite build",**

**"lint": "eslint . --ext js,jsx --report-unused-disable-directives --max-warnings 0",**

**"preview": "vite preview"**

**},**

**"dependencies": {**

**"@ant-design/icons": "^5.2.6",**

**"@reduxjs/toolkit": "^1.9.7",**

**"antd": "^5.11.2",**

**"axios": "^1.6.2",**

**"chart.js": "^4.4.8",**

**"html-react-parser": "^5.0.6",**

**"millify": "^6.1.0",**

**"moment": "^2.29.4",**

**"react": "^18.3.1",**

**"react-chartjs-2": "^5.2.0",**

**"react-dom": "^18.3.1",**

**"react-redux": "^8.1.3",**

**"react-router-dom": "^6.19.0",**

**"uuid": "^9.0.1"**

**},**

**"devDependencies": {**

**"@types/react": "^18.2.37",**

**"@types/react-dom": "^18.2.15",**

**"@vitejs/plugin-react": "^4.2.0",**

**"eslint": "^8.53.0",**

**"eslint-plugin-react": "^7.33.2",**

**"eslint-plugin-react-hooks": "^4.6.0",**

**"eslint-plugin-react-refresh": "^0.4.4",**

**"vite": "^5.4.14"**

**}**

**}**