



PROJECT REPORT

CRYPTOVERSE :A CRYPTOCURRENCY DASHBOARD

YEAR : 2024 – 2025

COLLEGE NAME : K.C.S KASI NADAR COLLEGE OF ARTS & SCIENCE

CODE : UNM203

DEPARTMENT : COMPUTER SCIENCE

PROGRAM : B.C.A

SEMESTER VI

PROJECT SUBMITTED TO: UNIVERSITY OF MADRAS / NAAN MUDALVAN

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

TEAM LEADER: K. Praveen

Email Id:praveenk2407chennai@gmail.com

MEMBERS:

1. S.Diwakar.

Email id : diwakarsenthilkumar11@gmail.com

2. B.Mohammed Fahimuddeen.

Email id : fahzxx5@gmail.com

3. L. Vijayakannan.

Email id : l.vijayakannankcs@gmail.com

GUIDED BY: MRS.R.PADMADEVI

SPOC NAME: Dr.K. LALITHAKAMESWARI

CRYPTOVERSE :A CRYPTOCURRENCY DASHBOARD

Introduction:-

As In recent years, cryptocurrencies have gained immense popularity, transforming the financial landscape by offering a decentralized and digital alternative to traditional banking. With the rise of Bitcoin, Ethereum, and thousands of altcoins, investors, traders, and enthusiasts require real-time data, historical analysis, and market insights to make informed decisions. However, tracking the price movements of multiple cryptocurrencies across various exchanges can be challenging due to the volatile and fast-paced nature of the market.

To address this challenge, Cryptoverse provides a feature-rich cryptocurrency dashboard that allows users to track live price updates, view historical trends, and analyze market performance through an interactive and user-friendly interface. The platform is designed for both beginners and experienced investors, offering essential tools for monitoring cryptocurrency prices, trends, and volatility.

Cryptoverse leverages React.js for a dynamic user interface, Redux Toolkit for efficient state management, and Chart.js for interactive data visualization. The application fetches real-time market data from an external API and presents it in a structured, easy-to-navigate format. Users can view global market statistics, search and filter cryptocurrencies, and access detailed coin information with just a few clicks.

One of the standout features of Cryptoverse is its interactive line charts, which allow users to analyze historical price trends for different timeframes. By visualizing past performance, users can identify patterns and make data-driven investment decisions. Additionally, the platform provides insights into market cap, trading volume, daily changes, and more, making it a comprehensive tool for crypto analysis.

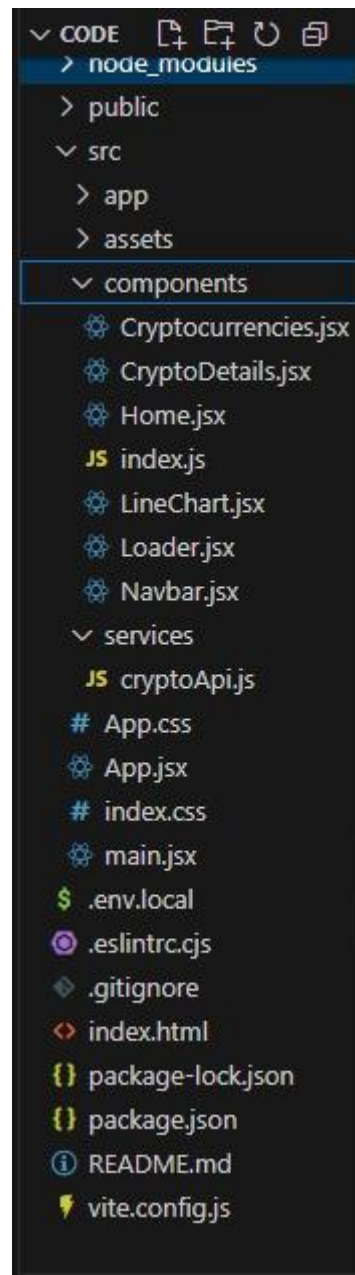
Key Features:-

- ❖ Historical Price Data - Track price movements over 5 years.
- ❖ Interactive Charts - View trends using React Chart.js.
- ❖ Search Functionality - Find specific cryptocurrencies easily.
- ❖ Real-Time Market Updates - Get the latest data with API integration.
- ❖ Compare Market Trends - Analyze performance between different cryptocurrencies.
- ❖ Educational Insights - Learn about market patterns & price influences.

Project Goals and Objectives:-

- ❖ To develop a real-time cryptocurrency dashboard that provides users with accurate market insights using interactive charts and secure API integration.
- ❖ Implement live market data updates with efficient state management.
- ❖ Ensure a user-friendly UI using React.js and Redux Toolkit.
- ❖ Integrate Chart.js for clear data visualization.
- ❖ Maintain security and performance optimization.
- ❖ Enable scalability for future feature enhancements.

Project structure:



Sample Code:

cryptoApi.js

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

const cryptoApiHeaders = {
  "X-RapidAPI-Key": import.meta.env.VITE_RAPID_API_KEY,
  "X-RapidAPI-Host": import.meta.env.VITE_RAPID_API_HOST,
};

const baseUrl = import.meta.env.VITE_BASE_URL;

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?timePeriod=${timePeriod}`),
    }),
  }),
});

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

Cryptocurrencies.jsx

```
Cryptocurrencies.jsx 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

```
CryptoDetails.jsximport 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: "Approved 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;
```

.env.local

```

.env.localVITE_RAPID_API_KEY=b9fd96cd71mshc7bc4bf3cae1144p19ded5jsn78f47fc5e36b
VITE_RAPID_API_HOST=coinranking1.p.rapidapi.com
VITE_BASE_URL=https://coinranking1.p.rapidapi.com

```

Home.jsx

```

Home.jsximport 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;

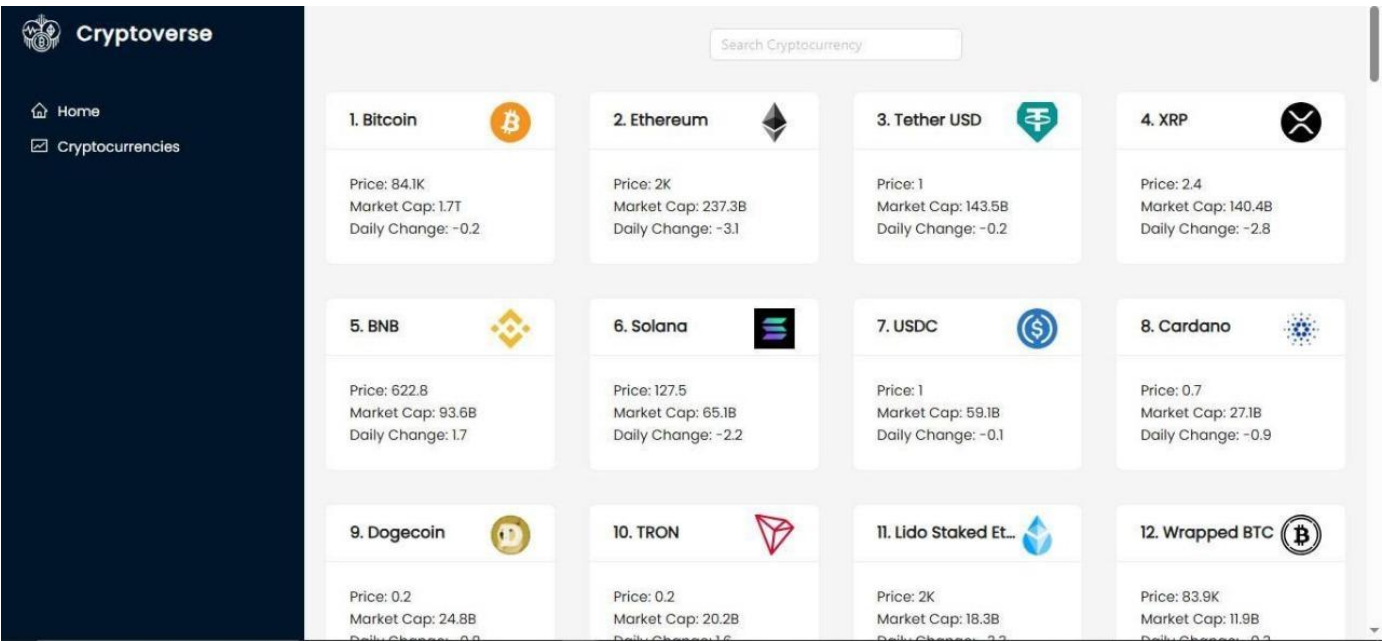
```

Project Execution:


❖ DASHBOARD



❖ CRYPTOCURRENCYS




❖ SEARCHING A CRYPTOCURRENCY

 Cryptoverse

Home

Cryptocurrencies

33. Monero



Price: 209.2

Market Cap: 3.9B

Daily Change: 2.3

mine

money

moneybags

monero

❖ LIVE STAT'S OF MONERO:



❖ VALUES AND ACCURATE STAT'S:

③

Price to USD

\$ 209.5

#

Rank

33

📈

24h Volume

\$ 66.1M

③

Market Cap

\$ 3.9B

📊

All-time-high(daily avg.)

\$ 520.3

What is Monero ?

Monero (XMR) is a private, untraceable cryptocurrency with dynamic scalability, ASIC resistance, and multiple keys.

Others statistics

An overview showing the stats of all cryptocurrencies

☑

Number Of Markets

36

📄

Number Of Exchanges

18

⌚

Approved Supply

✓

⌚

Total Supply

\$ 18.4M

⌚

Circulating Supply

\$ 18.4M

Monero Links

Website

getmonero.org

Bitcointalk

bitcointalk.org

Project Demo link :

<https://drive.google.com/file/d/17vXfpiRQdPs5bsTFCIOx-wGfJnV4-bfG/view?usp=drivesdk>

Project Source Code :

<https://github.com/ShadowsAreAlone/Cryptocurrency-.git>