**CRYPTOVERSE DASHBOARD**

**Team Details:**

**Team ID : SWTID1741324641146409**

Team Leader : Lokesh R – lokeshmt09.10@gmail.com

Team Member : Barathan V – barathanv12b107@gmail.com

Team Member : Kesavan P – kesavanp1309@gmail.com

Team Member : Santhosh V – sv067031@gmail.com

**INTRODUCTION**

The **Cryptoverse Dashboard** is an advanced React.js application designed to provide cryptocurrency enthusiasts and investors with real-time market insights and historical data analysis. Featuring a modern, dark-themed interface with neon accents, the dashboard offers a suite of powerful tools: global market statistics, a searchable and paginated list of cryptocurrencies, detailed coin profiles with interactive zoomable charts, and a comparison tool for side-by-side price analysis. Powered by the **CoinGecko API, Redux Toolkit for state management, and Chart.js for data visualization**, the **Cryptoverse Dashboard** empowers users to track trends, analyze historical data, and make informed investment decisions while serving as an educational resource.

**DESCRIPTION**

The **Cryptoverse Dashboard** is a comprehensive cryptocurrency tracking platform built with React.js. It integrates a sidebar navigation system, a homepage showcasing global crypto statistics and top performers, a paginated list of cryptocurrencies with search functionality, detailed coin pages with historical price charts, and a comparison feature with dual-line charts. The application leverages Redux Toolkit for efficient state management, React Router for seamless navigation, and Chart.js with zoom functionality for dynamic visualizations. Its responsive design, styled with a dark theme and neon highlights, ensures an engaging user experience across devices.

**SCENARIO**

Liam, a crypto trader, wants to evaluate market trends and compare investment options to refine his trading strategy.

1. **Objective**: Liam aims to identify cryptocurrencies with strong growth potential and compare their historical performance.
2. **Using Cryptoverse Dashboard**: Liam opens the app on his desktop.
3. **Navigation**: He uses the sidebar to switch between "Home," "All Currencies," and "Compare" pages.
4. **Global Overview**: On the homepage, Liam reviews global stats and the top 10 cryptocurrencies.
5. **Browsing Coins**: He navigates to "All Currencies," searches for "Solana," and browses the paginated list.
6. **Detailed Analysis**: Clicking Solana, Liam examines its 1-year price chart and key stats.
7. **Comparison**: On the "Compare" page, he selects Bitcoin and Solana to visualize their price trends.
8. **Decision-Making**: With actionable insights, Liam adjusts his portfolio allocation.

**PRE-REQUISITES**

* **Node.js and npm**:
  + Download: <https://nodejs.org/en/download/>
  + Install: <https://nodejs.org/en/download/package-manager/>
* **React.js**:
  + Install: npx create-react-app cryptoverse-dashboard
* **Libraries** (as specified in package.json):
  + @reduxjs/toolkit: "^2.6.0"
  + axios: "^1.8.1"
  + chart.js: "^4.4.8"
  + chartjs-plugin-zoom: "^2.2.0"
  + react: "^19.0.0"
  + react-chartjs-2: "^5.3.0"
  + react-dom: "^19.0.0"
  + react-redux: "^9.2.0"
  + react-router-dom: "^7.3.0"
  + react-scripts: "5.0.1"
  + Testing libraries: @testing-library/dom, @testing-library/jest-dom, @testing-library/react, @testing-library/user-event
  + Install: npm install
* **HTML, CSS, JavaScript**: Basic knowledge required.
* **Git**: Version control (<https://git-scm.com/downloads>).
* **Editor**: Visual Studio Code recommended (<https://code.visualstudio.com/>).

**PROJECT SETUP**

**Clone the Repository**

* **GitRepository**:(<https://github.com/Im-Lokesh-R/Cryptoverse-A-Cryptocurrency-Dashboard-SWTID1741324641146409>)
* **Command**:gitclone(<https://github.com/Im-Lokesh-R/Cryptoverse-A-Cryptocurrency-Dashboard-SWTID1741324641146409>)
* **Navigate**: cd cryptoverse-dashboard

**Install Dependencies**

* **Command**: npm install (Installs dependencies listed in package.json).

**Start the Development Server**

* **Command**: npm start (As defined in package.json scripts).
* **URL**: http://localhost:3000

**Verify Setup**

* Open http://localhost:3000 in your browser to see the dashboard with the React logo in the favicon.

**PROJECT STRUCTURE**

cryptoverse-dashboard/

├── public/

│ ├── index.html

│ ├── favicon.ico

│ ├── logo192.png

│ ├── logo512.png

│ ├── manifest.json

│ └── robots.txt

├── src/

│ ├── components/

│ │ ├── Home.jsx

│ │ ├── CryptoList.jsx

│ │ ├── CryptoDetails.jsx

│ │ ├── CryptoCompare.jsx

│ │ └── Navbar.jsx

│ ├── features/

│ │ └── cryptoSlice.js

│ ├── App.js

│ ├── App.css

│ ├── index.js

│ ├── index.css

│ ├── store.js

│ ├── reportWebVitals.js

├── .gitignore

├── package.json

├── package-lock.json

└── README.md

**PROJECT DEVELOPMENT**

**Root-Level Files**

* **.gitignore**:
  + Ignores: node\_modules, /build, .env files, logs (npm-debug.log\*, yarn-debug.log\*), and other system files (e.g., .DS\_Store).
* **package.json**:
  + Project metadata: Name (my-app), version (0.1.0), private (true).
  + Dependencies: Lists all required libraries (e.g., react, axios, chart.js).
  + Scripts: start, build, test, eject (standard Create React App scripts).
  + ESLint configuration: Extends react-app and react-app/jest.
  + Browserslist: Defines supported browsers for production and development.
* **package-lock.json**:
  + Locks dependency versions for reproducibility (e.g., react@19.0.0, axios@1.8.1).
* **README.md**:
  + Provides setup instructions, available scripts (npm start, npm test, npm run build, npm run eject), and links to Create React App documentation.

**Public Assets**

* **index.html**: The main HTML template for the React app, including meta tags for responsiveness and a favicon link to the React logo.
* **manifest.json**: Configures the app for progressive web app (PWA) functionality with icons (logo192.png, logo512.png) and a standalone display mode.
* **robots.txt**: Allows all user agents to crawl the site (Disallow: is empty).
* **favicon.ico**: The React logo (shown below) used as the app’s favicon.

**Reference Image (React Logo)**:

* The favicon (favicon.ico) displays the React logo, a blue atom-like symbol with orbiting paths, representing the React framework.

**Redux Store (store.js)**

* **Purpose**: Configures the Redux store for managing cryptocurrency data.
* **Code Breakdown**:
  1. import { configureStore } from '@reduxjs/toolkit';: Imports Redux Toolkit’s store utility.
  2. import cryptoReducer from './features/cryptoSlice';: Imports the crypto reducer.
  3. export default configureStore({...});: Exports a store with the crypto reducer.

**Crypto Slice (cryptoSlice.js)**

* **Purpose**: Defines async thunks and reducers for fetching and managing CoinGecko API data.
* **API Base URL**: https://api.coingecko.com/api/v3
* **Async Thunks**:
  1. fetchCryptoData: Fetches a paginated list of coins (/coins/markets).
  2. fetchCryptoDetails: Retrieves details for a specific coin (/coins/{coinId}).
  3. fetchCryptoHistory: Fetches historical price data (/coins/{coinId}/market\_chart).
  4. fetchGlobalData: Gets global market stats (/global).
  5. fetchCoinList: Fetches the full list of coins for dropdowns (/coins/list).
* **State**:
  + coins: Array of coin market data.
  + coinDetails: Object mapping coin IDs to details.
  + history: Object mapping coin IDs to price histories.
  + globalData: Global market stats.
  + coinList: Array of all coins for comparisons.
  + searchTerm: Filters coin list.

**App Component (App.js)**

* **Purpose**: Root component with routing and sidebar navigation.
* **Features**:
  1. **Routing**: Defines paths for Home, CryptoList, CryptoDetails, and CryptoCompare.
  2. **Sidebar**: Integrates search and navigation links styled with .sidebar.
  3. **State**: Fetches initial coin list and updates searchTerm via Redux.

**Home Component (Home.jsx)**

* **Purpose**: Displays global crypto stats and top 10 coins.
* **Features**:
  1. **Global Stats**: .global-stats grid with cards for market cap, volume, etc.
  2. **Top Coins**: .crypto-grid showing 10 coins with links to details.
  3. **Data Fetching**: Uses fetchCryptoData (12 coins) and fetchGlobalData.

**CryptoList Component (CryptoList.jsx)**

* **Purpose**: Lists all cryptocurrencies with search and pagination.
* **Features**:
  1. **Grid**: .crypto-grid displays 20 coins per page, filtered by searchTerm.
  2. **Pagination**: .pagination supports 50 pages with Previous/Next buttons.
  3. **Data Fetching**: Uses fetchCryptoData with dynamic page params.

**CryptoDetails Component (CryptoDetails.jsx)**

* **Purpose**: Shows detailed info and price history for a specific coin.
* **Features**:
  1. **Dynamic Route**: Uses coinId from useParams.
  2. **Chart**: Zoomable Line chart with time frame selector (24h to 5y).
  3. **Stats**: Displays price, market cap, supply, and 24h high/low.
  4. **Links**: External links with a styled .buy-link.

**CryptoCompare Component (CryptoCompare.jsx)**

* **Purpose**: Compares two coins with a dual-line price chart.
* **Features**:
  1. **Selection**: .selection-container with dropdowns for two coins and time frame.
  2. **Chart**: Dual-line chart comparing price histories with distinct colors.
  3. **Stats**: Current prices in .compare-stats.
  4. **Data Fetching**: Uses fetchCryptoDetails and fetchCryptoHistory.

**Navbar Component (Navbar.jsx)**

* **Purpose**: Sidebar navigation with search functionality.
* **Features**:
  1. **Search**: Updates searchTerm in Redux state.
  2. **Links**: NavLinks to Home and All Currencies with .active styling.

**Styling (App.css)**

* **Purpose**: Defines a dark theme with neon accents.
* **Highlights**:
  + .sidebar: Fixed 250px width with #00d4ff and #ff007a colors.
  + .crypto-grid: Responsive grid with minmax(220px, 1fr).
  + .card: Hover scaling and shadow effects.

**PROJECT EXECUTION**

**Demo Link**

* (<https://drive.google.com/file/d/1Ysy0K9Pe0rJXl-kuNCD0g3LIMNLFg21P/view?usp=drive_link>)

**GitHub Link**

* (<https://github.com/Im-Lokesh-R/Cryptoverse-A-Cryptocurrency-Dashboard-SWTID1741324641146409>)

**User Interface Snippets**

1. **Home Page**: Displays global crypto stats (e.g., total market cap, 24h volume, active cryptocurrencies) and a grid of the top 10 coins with their price, 24h change, and market cap.



1. **All Currencies Page**: A paginated list of cryptocurrencies with a search bar, displaying 20 coins per page in a responsive grid.



1. **Crypto Details Page**: Detailed view of a selected coin, featuring a zoomable price chart (24h to 5y), current price, market cap, supply stats, and external links (e.g., website, buy link).



1. **Compare Page**: Side-by-side comparison of two coins with dropdown selection, a dual-line price chart, and current price stats.



**PROJECT FLOW**

1. **Setup**:
   * Create React app: npx create-react-app cryptoverse-dashboard.
   * Install dependencies: npm install (as per package.json).
   * Configure store and routes.
2. **Design UI**:
   * Build sidebar and main content layout in App.js.
   * Style with App.css for dark theme and responsiveness.
3. **Implement Logic**:
   * Fetch data via CoinGecko API in cryptoSlice.js.
   * Bind data to components with Redux and render charts with Chart.js.

**NOTES**

* **API Details**: The app uses the CoinGecko API (https://api.coingecko.com/api/v3) for free, real-time cryptocurrency data. No API key is required, but rate limits apply (consult [CoinGecko API docs](https://www.coingecko.com/en/api/documentation)).
* **Testing**: The project includes testing libraries (@testing-library/jest-dom, etc.) and can be tested with npm test (as per README.md).
* **Enhancements**: The app can be extended with features like user authentication, portfolio tracking, or additional chart types.

**REFERENCES**

* **Create React App Documentation**: <https://facebook.github.io/create-react-app/docs/getting-started>
* **React Documentation**: <https://reactjs.org/>
* **CoinGecko API Documentation**: <https://www.coingecko.com/en/api/documentation>
* **Redux Toolkit Documentation**: <https://redux-toolkit.js.org/>
* **Chart.js Documentation**: <https://www.chartjs.org/docs/latest/>
* **React Router Documentation**: <https://reactrouter.com/>

**CONCLUSION**

The **Cryptoverse Dashboard** represents a robust and user-friendly solution for cryptocurrency enthusiasts and traders, leveraging modern web technologies to deliver real-time data and insightful visualizations. Through its integration of the **CoinGecko API, Redux Toolkit, and Chart.js**, the application provides a scalable foundation that can be further enhanced with additional features such as portfolio management or real-time alerts. The project’s responsive design and intuitive interface make it accessible across devices, while the inclusion of testing tools ensures maintainability. As a collaborative effort, this dashboard not only meets current market analysis needs but also sets the stage for future development in the rapidly evolving crypto space. With the provided GitHub repository, demo, and video, users and contributors can explore, deploy, and extend this project to suit their needs.