|  |
| --- |
| **PONNUSAMY NADAR COLLEGE OF ARTS AND SCIENCE**  **C.T.H ROAD, THOZHUVUR-THIRUVALLUR (DISTRICT)**      **CRYPTO CURRENCY DASHBIARD**    **TEAM HEAD :**    **S.AKASH**    **TEAM MEMBERS :**  **S,ANZARI**  **K.EZHLIARASAN**  **V.GOPI**  **Introduction:**    A crypto currency dashboard that displays historical price data over the past five years is a powerful tool for investors seeking a comprehensive understanding of market dynamics. This feature-rich interface offers users a detailed historical perspective on the performance of various crypto currencies, enabling insightful analysis and informed decision-making. Through visually intuitive charts and graphs, the dashboard allows for effective comparisons of multiple crypto currencies, aiding in the identification of top performers and overall market trends. Users can customize timeframes for a more granular examination of price movements, facilitating in-depth volatility analysis and risk assessment. This historical data not only supports investors in making data-driven decisions but also assists in recognizing recurring patterns and cycles. Beyond its role in optimizing crypto currency portfolios, the dashboard serves as an educational resource, empowering users to grasp the evolving nature of crypto currency markets and the nuanced factors shaping price movements over an extended period. Description  **Description :**  Crypto verse is a sophisticated crypto currency dashboard designed to provide investors with comprehensive insights into market dynamics through detailed historical price data analysis spanning five years. Featuring visually intuitive charts, interactive tools, and seamless navigation, the platform empowers users to identify top-performing assets and make informed investment decisions. With its robust search functionality, users can easily explore a wide range of crypto currencies and compare their performance over time. Cryptoverse not only serves as a powerful tool for optimizing investment portfolios but also acts as an educational resource, helping users understand the evolving nature of crypto currency markets  **CODING:**  {  "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": "^4.4.0",  "html-react-parser": "^5.0.6",  "millify": "^6.1.0",  "moment": "^2.29.4",  "react": "^18.2.0",  "react-chartjs-2": "^5.2.0",  "react-dom": "^18.2.0",  "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": "^5.0.0"  }  }  **CODING EXPLANATION :**  **1.BASIC INFORMATION**  The package.json file provided contains essential metadata and configuration for a JavaScript project named "crypto." The project is marked as private, meaning it is not intended to be published to npm (Node Package Manager), and its version is currently set to "0.0.0", which is likely a placeholder for initial development. The type field is set to "module", indicating that the project uses ECMAScript Modules (ESM) for importing and exporting files, as opposed to the older CommonJS syntax. This setup is typical for modern JavaScript projects, especially those leveraging tools like Vite and React. The file also includes scripts for running development servers, building the project, linting code, and previewing production builds, making it a comprehensive configuration for managing the project's lifecycle.  **2.SCIPTS:**  The scripts section in the package.json file defines a set of commands that can be executed to perform various tasks in the project. These scripts are shortcuts that simplify common development workflows. For example, the dev script runs the development server using vite, allowing you to see changes in real-time as you code. The build script uses vite build to create an optimized production-ready version of the project. The lint script runs ESLint to check the code for errors, enforce coding standards, and ensure consistency across the project. Lastly, the preview script uses vite preview to locally serve the production build, allowing you to test how the final version of the app will behave before deploying it. These scripts streamline tasks like development, testing, and deployment, making the process more efficient.  **3.DEPENDENCIES:**  The dependencies section in the package.json file lists all the external libraries and packages that the project needs to function properly. These are essential for the application to run in production. For example, react and react-dom are core libraries for building the user interface, while axios is used for making HTTP requests to APIs. Libraries like antd and @ant-design/icons provide pre-built UI components and icons, making it easier to design the application. State management is handled by @reduxjs/toolkit and react-redux, while react-router-dom enables navigation between different pages in the app. Additional utilities like chart.js and react-chartjs-2 are used for data visualization, millify simplifies large numbers, and uuid generates unique identifiers. These dependencies work together to provide the functionality needed for the project to operate as intended.  **4. Dev DEPENDENCIES:**  The dev Dependencies section in the package.json file lists packages that are only needed during development and not in the final production build. These tools help developers write, test, and maintain the code more effectively. For example, @types/react and @types/react-dom provide TypeScript type definitions for React, ensuring better code safety and autocompletion. The @vitejs/plugin-react plugin enables Vite to work seamlessly with React. Tools like eslint and its associated plugins (eslint-plugin-react, eslint-plugin-react-hooks, and eslint-plugin-react-refresh) are used to lint the code, catching errors and enforcing coding standards. Finally, vite itself is a fast build tool that powers the development server, builds the project, and enables previewing. These dev dependencies streamline the development process, improve code quality, and enhance productivity.    **CONCLUSION:**  The cryptocurrency project is a modern web application built using React and powered by Vite for fast development and production builds. It leverages popular libraries like Ant Design for UI components, Redux for state management, and Chart.js for data visualization, making it feature-rich and user-friendly. The project also integrates tools like Axios for API communication, React Router for navigation, and ESLint for maintaining code quality. With a focus on scalability and performance, this project is well-equipped to handle real-time cryptocurrency data, provide interactive charts, and deliver a seamless user experience. Whether for tracking market trends or analyzing crypto data, this application serves as a robust and efficient solution for cryptocurrency enthusiasts and developers alike. |