**FRONTEND DEVELOPMENT WITH REACT.JS PROJECT DOCUMENTATION**

1. **INTRODUCTION:-**

**PROJECT TITLE: CRYPTO CURRENCY DASHBOARD**

**TEAM MEMBERS: 4**

**TEAM LEADER: H PAVITHRA**

**TEAM MEMBER 1. M YOGALAKSHNI**

**2. M SANGEETHA**

**3. A PRIYA**

1. **PROJECT OVERVIEW:-**

A Cryptocurrency Dashboard is a tool designed to provide users with real-time information and analytics about various cryptocurrencies.

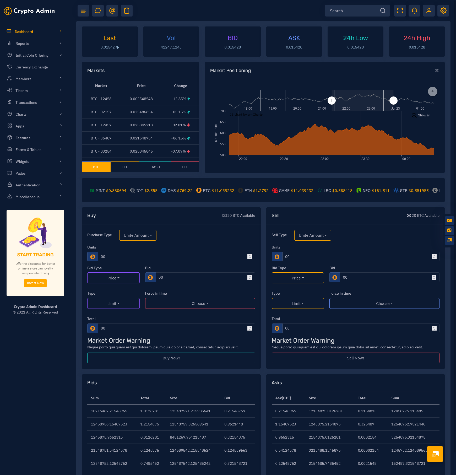
**Purpose:**

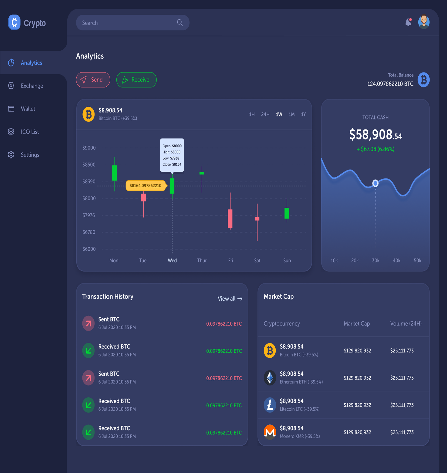
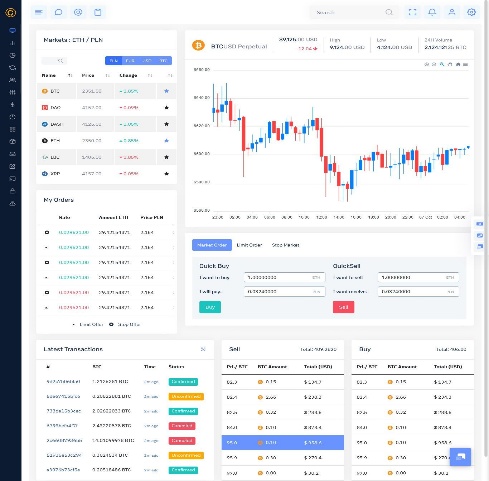
* To track the performance and price trends of different cryptocurrencies.
* To help users make informed decisions about buying, selling, or holding assets.
* To provide insights and analysis of the crypto market.

**Features:**

* Price Tracking: Real-time updates of cryptocurrency prices across multiple exchanges.
* Portfolio Management: Users can track their investments, monitor gains/losses, and portfolio performance.
* Market Analysis: Charts, graphs, and technical indicators to analyze trends.
* News Feed: Latest news and updates related to cryptocurrencies.
* Alerts/Notifications: Customizable alerts for price changes or market events.
* Wallet Integration: Allows users to connect their wallets to track balances.

1. **ARCHITECTURE:-**





**A Cryptocurrency Dashboard serves as a centralized platform to monitor and manage cryptocurrency investments, providing real-time data and analytics. Here's a concise overview of its architecture:**

**1. Component Structure:**

* **Frontend:**
  + **User Interface (UI):** Developed using frameworks like React or Angular, the UI presents real-time data, charts, and user interactions.
  + **State Management**: Utilizes tools such as Redux or React's Context API to manage the application's state, ensuring consistent data flow and responsiveness.

[**upgrad.com**](https://www.upgrad.com/blog/react-js-architecture/?utm_source=chatgpt.com)

* **Backend:**
  + **Server:** Built with technologies like Node.js or Django, it handles API requests, user authentication, and data processing.
  + Database: Stores user data, transaction histories, and other persistent information using systems like MongoDB or PostgreSQL.
* External Services:
  + Cryptocurrency APIs: Fetch real-time market data from sources like CoinGecko or Coinbase.

[coingecko.com](https://www.coingecko.com/learn/crypto-price-tracker-react?utm_source=chatgpt.com)

* + Authentication Services: Integrate with platforms like Web3Auth for secure user authentication.

[blog.web3auth.io](https://blog.web3auth.io/intents-blockchain-architecture/?utm_source=chatgpt.com)

**2. State Management:**

* **Frontend State:**
  + **Local State:** Managed within individual components for transient data.
  + **Global State:** Utilizes state management libraries to handle data shared across multiple components, ensuring synchronization and scalability.
* **Backend State:**
  + **Session Management:** Maintains user sessions and authentication tokens.
  + **Data Consistency:** Ensures reliable transactions and data integrity, especially during high-frequency data updates.

**3. Routing:**

* **Frontend Routing:**
  + Client-Side Routing: Employs libraries like React Router to navigate between different views (e.g., dashboard, portfolio, settings) without full page reloads, enhancing user experience**.**
* **Backend Routing:**
  + API Endpoints: Defines routes to handle various HTTP requests (GET, POST, etc.) for data retrieval, submission, and user actions.

1. **SET UP INSTRUCTIONS:**

Prerequisites:

-Node.js

-npm or yam

-React.js

-Redux toolkit

1. **Installation**:

1. Clone repository: git clone <https://github.com/your-repo/cryptoverse.git>

2. Install dependencies: npm install

3. Configure environment variables

1. **Folder Structure**

/cryptoverse

??? /src ?

??? /components

? ??? /pages

? ??? /redux

? ??? /utils

? ??? /assets

??? package.json

??? README.md

1. **Running the Application**

To start development server:

npm start

1. **Component Documentation Key Components:**

-CryptoCard: Cryptocurrency overview

- PriceChart: Historical price data

- NewsFeed: Crypto news

1. **Reusable Components:**

- Button: Customizable

- Loader: Loading animations

**10. State Management**

Global State: Managed using Redux Toolkit

Local State: Managed with React?s useState

**11.Styling**

Using Tailwind CSS with a custom theme for a modern UI

**12.Testing**

Testing Strategy:

- Unit Testing: Jest & React Testing Library

- Integration Testing: Cypress for E2E tests

**13.Known Issues**

API rate limits may affect updates.

Dark mode support in progress.

14.**Future Enhancements Future Plans:**

- Push notifications for price alerts

- AI-driven price predictions

- Staking feature for rewards