Frontend Development with React.js

1. Introduction

Project Title: CRYPTOCURRENCY DASHBOARD

Team Members:

|  |  |  |
| --- | --- | --- |
| **TEAM MEMBER** | **NAME** | **EMAIL** |
| TEAM LEADER | LAVANYA SREE K. | [lavanyasreekarunanithi@gmail.com](mailto:lavanyasreekarunanithi@gmail.com) |
| TEAM MEMBER | BRINDHA R. | [brindha8205@gmail.com](mailto:brindha8205@gmail.com) |
| TEAM MEMBER | PRIYADHARSHINI B. | [priyadharshinibanubabu@gmail.com](mailto:priyadharshinibanubabu@gmail.com) |
| TEAM MEMBER | KAVIN MALAR A.V. | [vvalarmathi170@gmail.com](mailto:vvalarmathi170@gmail.com) |
| TEAM MEMBER | SANTHOSH N. | [santhoshluckyjam@gmail.com](mailto:santhoshluckyjam@gmail.com) |

2. Project Overview

Purpose:

The Cryptocurrency Dashboard provides real-time updates and visual analytics on various cryptocurrencies. It aims to deliver a user-friendly platform for tracking market trends, price fluctuations, and historical data through an interactive interface. The dashboard simplifies cryptocurrency analysis by presenting daily updates in an easy-to-understand visual format.

Features:

* Real-Time Data Updates: Live cryptocurrency price updates with historical trends.
* Interactive Charts & Graphs: Visual representation of market data for better analysis.
* Portfolio Management: Users can track their investments and monitor gains/losses.
* Decentralization: Operates on blockchain technology, ensuring transparency and security.
* Security: Advanced cryptographic techniques secure transactions and prevent fraud.
* User-Friendly UI: Simplified layout for better accessibility and understanding.

3. Architecture

Component Structure:

* Overview of React components and their responsibilities.

State Management:

* Description of the approach used (e.g., Context API, Redux).

Routing:

* Explanation of routing structure if using React Router or other libraries.

4. Setup Instructions

Prerequisites:

* Node.js (latest stable version)

Installation:

1. git link <https://github.com/Lav-ab/Naan-Mudhalvan-Project>
2. Navigate to the project directory: cd cryptocurrency-dashboard
3. Install dependencies: npm install
4. Configure environment variables if necessary.

5. Folder Structure

Client:

Utilities:

* Explanation of helper functions, utility classes, or custom hooks.

6. Running the Application

Commands:

* Start frontend server locally: npm start

7. Component Documentation

Key Components:

* Dashboard: Main interface displaying real-time cryptocurrency data and charts.
* Navbar: Navigation bar allowing users to switch between different sections.
* CryptoList: Component displaying a list of top cryptocurrencies with key metrics.
* CryptoDetails: Page providing detailed information about a selected cryptocurrency.
* ChartComponent: Interactive chart visualizing historical price trends.
* SearchBar: Enables users to search for specific cryptocurrencies.
* PortfolioTracker: Allows users to track their investments and calculate profit/loss.
* ThemeSwitcher: Toggles between light and dark mode for better accessibility.
* ErrorBoundary: Handles application errors and provides fallback UI.

Reusable Components:

* Button: Customizable button component used across the application.
* Card: Generic card component for displaying cryptocurrency details.
* Loader: Loading animation used during data fetching.
* Modal: Popup modal used for alerts and additional information.

8. State Management

Global State:

* How state is managed across the application using Redux or Context API.

Local State:

* Handling of local states within individual components.

9. User Interface

* Screenshots or GIFs showcasing different UI features (charts, graphs, portfolio tracker, etc.).

10. Styling

CSS Frameworks/Libraries:

* Details of CSS frameworks or libraries used (e.g., Tailwind CSS, Styled-Components).

Theming:

* Explanation of theming or custom design systems implemented.

11. Testing

Testing Strategy:

* Unit, integration, and end-to-end testing approach (e.g., Jest, React Testing Library).

Code Coverage:

* Tools or techniques used for ensuring adequate test coverage.

12. Screenshots or Demo

* <https://drive.google.com/drive/folders/12aog4RuaOVBa1cHVPH_IS_Y1Rdek1dIZ>

13. Future Enhancements

* Outline potential future features, improvements, new components, animations, or enhanced styling, such as AI-based predictions or automated alerts.