PROJECT DOCUMENTATION FOR CRYPTOCURRENCY

1. Introduction

- Project Title: Cryptocurrency Tracker
- Team Members:
  - [Member Name :PRASANNA KUMAR SK] - Role Team lead
  - [Member Name 2:BALAJI D] - Role Team member
  - [Member Name 3:JESWIN PRAISON S] - Role Team member
  - [Member Name 4:KAMALESHWARAN] - Role Team member
  - [Member Name 5:SYED NIZAMUDDIN] - Role Team member

2. Project Overview

- Purpose:
  - The purpose of this project is to create a user-friendly application that allows users to track cryptocurrency prices and trends in real-time.
- Features:
  - Real-time price updates
  - Historical data visualization
  - User authentication
  - Portfolio management

3. Architecture

- Component Structure:
  - The application is structured into several major components including Home, CryptoDetails, Cryptocurrencies, and LineChart.
- State Management:
  - The project uses Redux for global state management to handle the application state efficiently.
- Routing:

  - React Router is used for navigation between different pages of the application.

4. Setup Instructions

- Prerequisites:
  1. Node.js
  2. npm (Node Package Manager)
- Installation:
  1. Clone the repository: `git clone https://github.com/Bycks/cryptocurrency.git`
  2. Navigate to the project directory: `cd cryptocurrency`
  3. Install dependencies: `npm install`
  4. Configure environment variables as needed.

5. Folder Structure

- Client:
  - Contains the React application organized into folders such as components, pages, and assets.
- Utilities:
  - Includes helper functions and custom hooks used throughout the project.

6. Running the Application

- Commands:
  - To start the frontend server locally, run: `npm start` in the client directory.

7. Component Documentation

- Key Components:
  - Home: Displays the main dashboard with cryptocurrency prices.

  - CryptoDetails: Shows detailed information about a selected cryptocurrency.
  - Cryptocurrencies: Lists all available cryptocurrencies.
  - LineChart: Visualizes historical price data.
- Reusable Components:
  - Loader: Displays a loading spinner while data is being fetched.
  - Navbar: Navigation bar for the application.

8. State Management

- Global State:
  - Managed using Redux, allowing for a centralized store for all application state.
- Local State:
  - Handled within individual components using React's useState hook.

9. User Interface

- Screenshot:
  - Include screenshots or GIFs showcasing different UI features, such as the homepage, cryptocurrency details page, and charts.

10. Styling

- CSS Frameworks/Libraries:
  - The project uses CSS modules for styling components.
- Theming:
  - Custom theming is implemented to provide a consistent look and feel across the application.

11. Testing

- Testing Strategy:
    - The project employs unit testing using Jest and React Testing Library to ensure component functionality.
- Code Coverage:
    - Code coverage is monitored using Jest's built-in coverage tool.

12. Screenshots or Demo

- Demo Link:
    - Provide a link to a live demo or screenshots showcasing the application’s features and design.

13. Known Issues

- Documented Issues:
    - List any known bugs or issues that users or developers should be aware of.

14. Future Enhancements

- Potential Features:
    - Integration of additional cryptocurrencies.
    - Enhanced data visualization options.
    - User notifications for price changes