**CRYPTOVERSE – A Cryptocurrency Dashboard**

**1. Introduction**

* **Project Title:** Crypto Dashboard
* **Team Leader:** Hemalatha S [mail id: sivamukanhema9@gmail.com ]
* **Team Members:**
  + Archana P [mail id: archanaponnusamy43@gmail.com ]
  + Archana S [mail id: archanaarchana89120@gmail.com ]
  + Dhanalakshmi A[mail id: dhanalakshmipriya27@gmail.com ]
  + Divya B [mail id: db8111468@gmail.com ]

**2. Project Overview**

* **Purpose:**  
  The Crypto Dashboard allows users to track real-time cryptocurrency prices, view detailed market trends, and monitor their portfolio performance. The application provides an interactive and visually appealing interface for users to manage and analyze crypto assets.
* **Features:**
  + Real-time cryptocurrency prices and market data.
  + Portfolio management with custom asset tracking.
  + Charts and graphs for market trends.
  + Search functionality for coins and tokens.
  + Interactive notifications for price alerts.

**3. Architecture**

* **Component Structure:**
  + **App Component**: The root component that contains routes and global states.
  + **Dashboard Component**: Displays the main cryptocurrency data.
  + **Portfolio Component**: Manages and displays user portfolio.
  + **CoinDetail Component**: Provides detailed information on individual coins.
* **State Management:**
  + The project uses **React Context API** for state management to handle global state such as user preferences, portfolio data, and coin data.
* **Routing:**
  + React Router is used for routing between different pages like Home, Portfolio, and Coin Details.

**4. Setup Instructions**

* **Prerequisites:**
  + Node.js
  + npm
* **Installation:**
  + Clone the repository: git clone

https://github.com/shema0425/Crypto-DashBoard-app

* + Navigate to the project directory: cd Crypto-Dashboard-app
  + Install dependencies: npm install
  + Configure environment variables

**5. Folder Structure**

* **Client:**
  + **src/components**: Contains reusable components like CoinCard, Header, and SearchBar.
  + **src/pages**: Contains pages like Dashboard, Portfolio, and CoinDetails.
  + **src/assets**: Stores images, icons, and other assets.
* **Utilities:**
  + Custom hooks like useFetchData for fetching API data.
  + Helper functions for formatting currency and managing portfolio.

**6. Running the Application**

* **Frontend:**  
  To start the frontend server locally:
  + Run the command: npm start in the client directory.

**7. Component Documentation**

* **Key Components:**
  + **CoinCard**: Displays individual coin data with price and market details.
    - **Props:** coinData (object)
  + **Portfolio**: Displays the user’s cryptocurrency portfolio.
    - **Props:** portfolioData (array)
* **Reusable Components:**
  + **Header**: Contains navigation and search functionality.
    - **Props:** onSearch (function)

**8. State Management**

* **Global State:**
  + The global state is managed using the Context API to store and share data like portfolio, user preferences, and active coins across the app.
* **Local State:**
  + Local state is handled within components using React’s useState hook, especially for user input and temporary data like search results or coin-specific details.

**9. User Interface**

* **Screenshots or GIFs:**
  + Dashboard with live prices.
  + Portfolio tracking page.
  + Interactive charts displaying market trends.

**10. Styling**

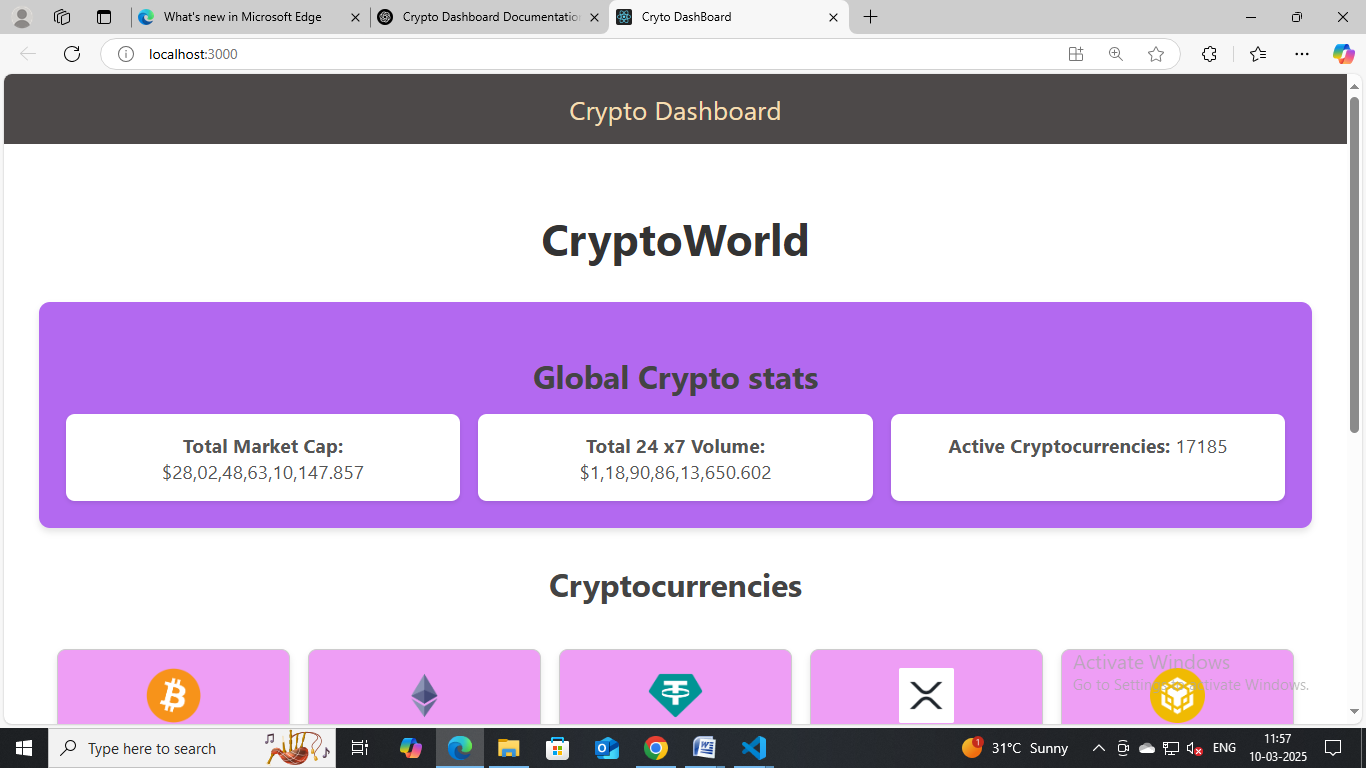
* **CSS Frameworks/Libraries:**
  + **Styled-Components** for styled elements.
  + **CSS Grid** and **Flexbox** for layout.
* **Theming:**
  + The application implements a light/dark theme toggle, allowing users to switch between themes for better accessibility.

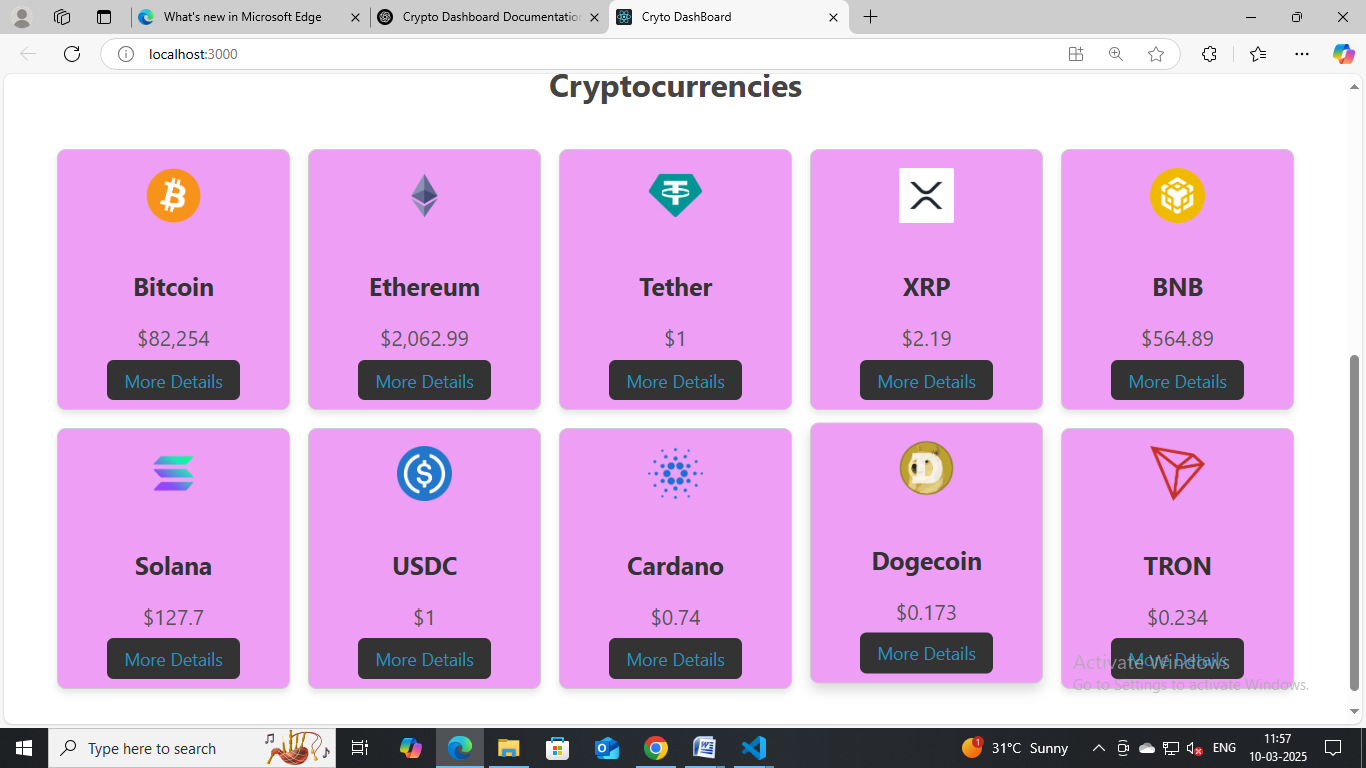
**11. Testing**

* **Testing Strategy:**
  + Unit tests for components using **Jest**.
  + Integration tests using **React Testing Library**.
* **Code Coverage:**
  + **Jest** is used to ensure sufficient test coverage for components and utilities.

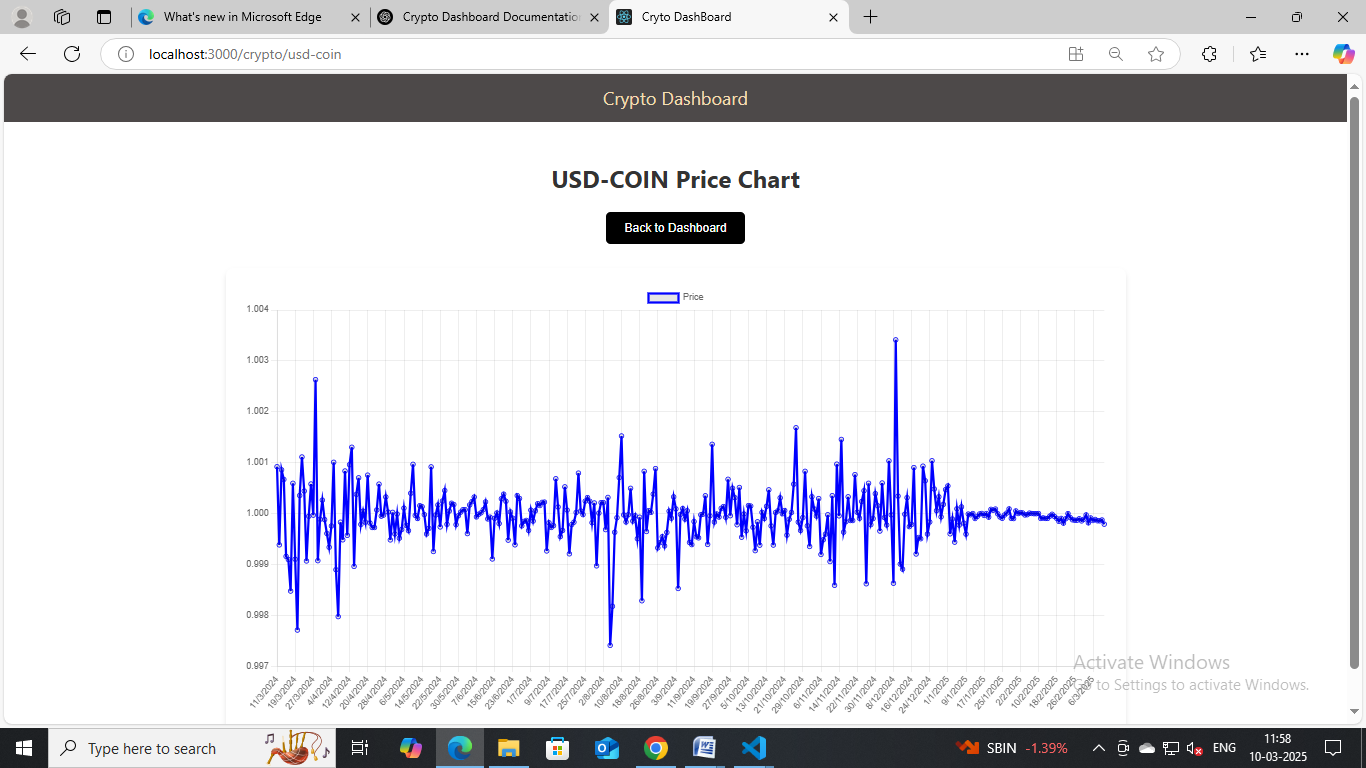
**12. Screenshots or Demo**

* **Screenshots:**

****

****

****

****

* **Demo video link**
* Link to demo:

https://drive.google.com/file/d/1vcJINR7X4pC42pMBbEHur5oSkYt0SfTf/view?usp=drivesdk

**13. Known Issues**

* Some real-time data might occasionally have delays due to API rate limits.
* Portfolio data might not sync instantly across devices.

**14. Future Enhancements**

* Adding real-time notifications for price changes.
* Implementing dark mode for improved user experience.
* Adding more advanced charts and analysis features for power users.