**Frontend Development with React.js**

**Project Documentation for Cryptocurrency Dashboard**

**1.Introduction**

* **Project Title : Cryptocurrency Dashboard**
* **Team Members :**
* Logeshwari S (Team Leader)

[Email Id:logeshwaris859@gmail.com]

* Lakshmi Priya K

[Email Id:lakshmipriyakannan24@gmail.com]

* Madumitha T

[Email Id: madhumitha29102004@gmail.com]

* Magisha T

[Email Id: annalub07@gmail.com]

**2. Project Overview**

* **Purpose :**

The primary purpose of Cryptoverse is to:

* Provide historical price data for cryptocurrencies over the past five years.
* Enable users to compare multiple cryptocurrencies and identify market trends.
* Serve as an educational resource for understanding cryptocurrency markets.
* **Features :**
* Historical Price Data: Displays price trends over customizable timeframes.
* Interactive Charts: Powered by `react-chartjs-2` and `Chart.js`.
* Search Functionality: Allows users to search for specific cryptocurrencies.
* Educational Resource: Helps users understand market dynamics and price movements.

**3. Architecture :**

* **Component Structure**

The application is built using “React.js” and follows a component-based architecture. Key components include:

* **Home:** Displays global cryptocurrency statistics and top 10 cryptocurrencies.
* **Cryptocurrencies:** Lists all cryptocurrencies with search functionality.
* **CryptoDetails:** Shows detailed information about a specific cryptocurrency.
* **LineChart:** Renders historical price data in a line chart.
* **Loader:** Displays a loading indicator while data is being fetched.
* **State Management :**
* **Global State**: Managed using “Redux Toolkit” for shared data across components.
* **Local State**: Managed using React's `useState` for component-specific data (e.g., search terms, time periods).

**4. Routing :**

* **React Router:** Used for client-side routing, enabling seamless navigation between pages.

**5.Setup Instructions:**

* **Prerequisites:**
* **Node.js and npm**: Required to run JavaScript on the server-side.
* **Git**: For version control and collaboration.
* **React.js**: JavaScript library for building user interfaces.**Git**: For version control and collaboration.
* **Code Editor**: Such as Visual Studio Code, Sublime Text, or WebStorm.
* **Installation :**

**1. Clone the Repository:** git clone <https://github.com/SSC369/cryptoverse>

**2. Navigate to the Project Directory:** cd cryptoverse

**3. Install Dependencies:** npm install

**4. Start the Development Server:** npm start

**5. Access the Application:** Open your web browser and navigate to `http://localhost:3000`.

**5. Folder Structure:**

* **Client**

**src:** Contains the main application code.

**app:** Redux store configuration.

**assets:** Static assets like images.

**components:** React components.

**services:** API service configurations.

**App.css:** Global styles.

**App.jsx:** Main application component.

**main.jsx:** Entry point for the application.

* **Utilities**

**eslintrc.js:** ESLint configuration.

**.gitignore:** Specifies files to be ignored by Git.

**package.json:** Lists project dependencies and scripts.

**6. Running the Application :**

To run the application, execute the following command in the project directory:

npm start

The application will be accessible at `http://localhost:3000`.

**7. Component Documentation :**

* **Key Components :**

**Home:** Displays global stats and top 10 cryptocurrencies.

**Cryptocurrencies:** Lists all cryptocurrencies with search functionality.

**CryptoDetails:** Shows detailed information about a specific cryptocurrency.

**LineChart:** Renders historical price data in a line chart.

**Loader:** Displays a loading indicator.

* **Reusable Components :**

**Card:** Used to display cryptocurrency information in a grid layout.

**Statistic:** Displays key statistics in a formatted manner.

**7. State Management :**

* **Global State:**

**Toolkit:** Manages shared data (e.g., API responses) across components.

**API Integration:** Handles cryptocurrency data fetched from external APIs.

**Redux Middleware:** Manages asynchronous API requests.

* **Local State :**

**React's `useState`:** Manages component-specific data (e.g., search terms, time periods).

**Example:** Redux Store Configuration

```javascript

import { configureStore } from "@reduxjs/toolkit";

import { cryptoApi } from "../services/cryptoApi";

export default configureStore({

    reducer: {

        [cryptoApi.reducerPath]: cryptoApi.reducer,

    },

    middleware: (getDefaultMiddleware) =>

        getDefaultMiddleware().concat(cryptoApi.middleware),

});

```

**9. User Interface :**

* **Home Page :**Displays global cryptocurrency statistics (e.g., total cryptocurrencies, total exchanges, market cap).

Lists the top 10 cryptocurrencies with key metrics.

* **Cryptocurrencies Page :**Lists all cryptocurrencies with search functionality.

Allows users to filter and search for specific cryptocurrencies.

* **Crypto Details Page :**Displays detailed information about a specific cryptocurrency, including historical price data and statistics.

**10. Styling :**

**CSS:** Custom styles for components.

**Ant Design:** Pre-built components for a cohesive and visually appealing interface.

**11. Testing :**

* **Testing Strategies :**

**Functionality Testing:** Ensure all features work as expected.

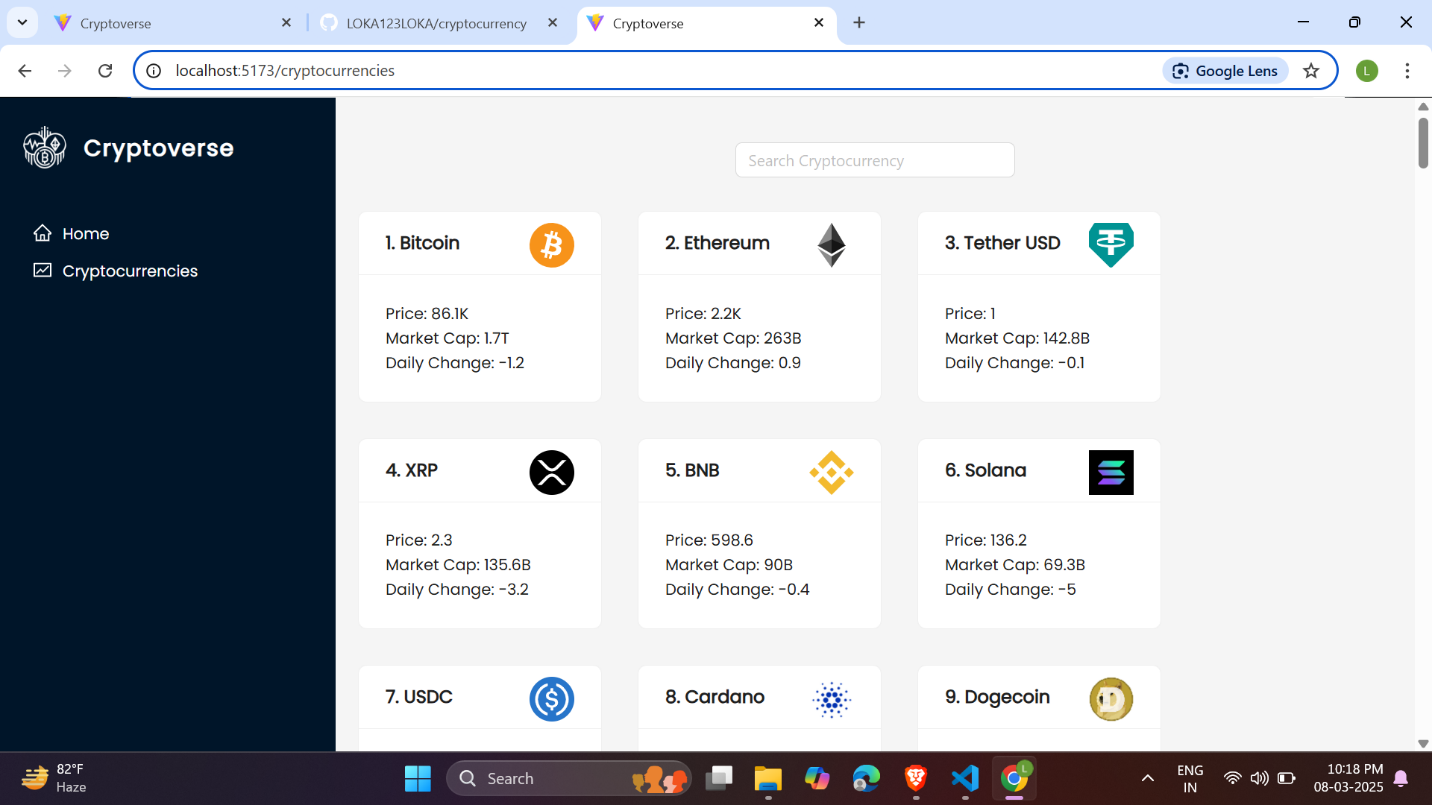
**Performance Testing:** Check for performance bottlenecks.

**Responsiveness Testing:** Ensure the application works well on different screen sizes.

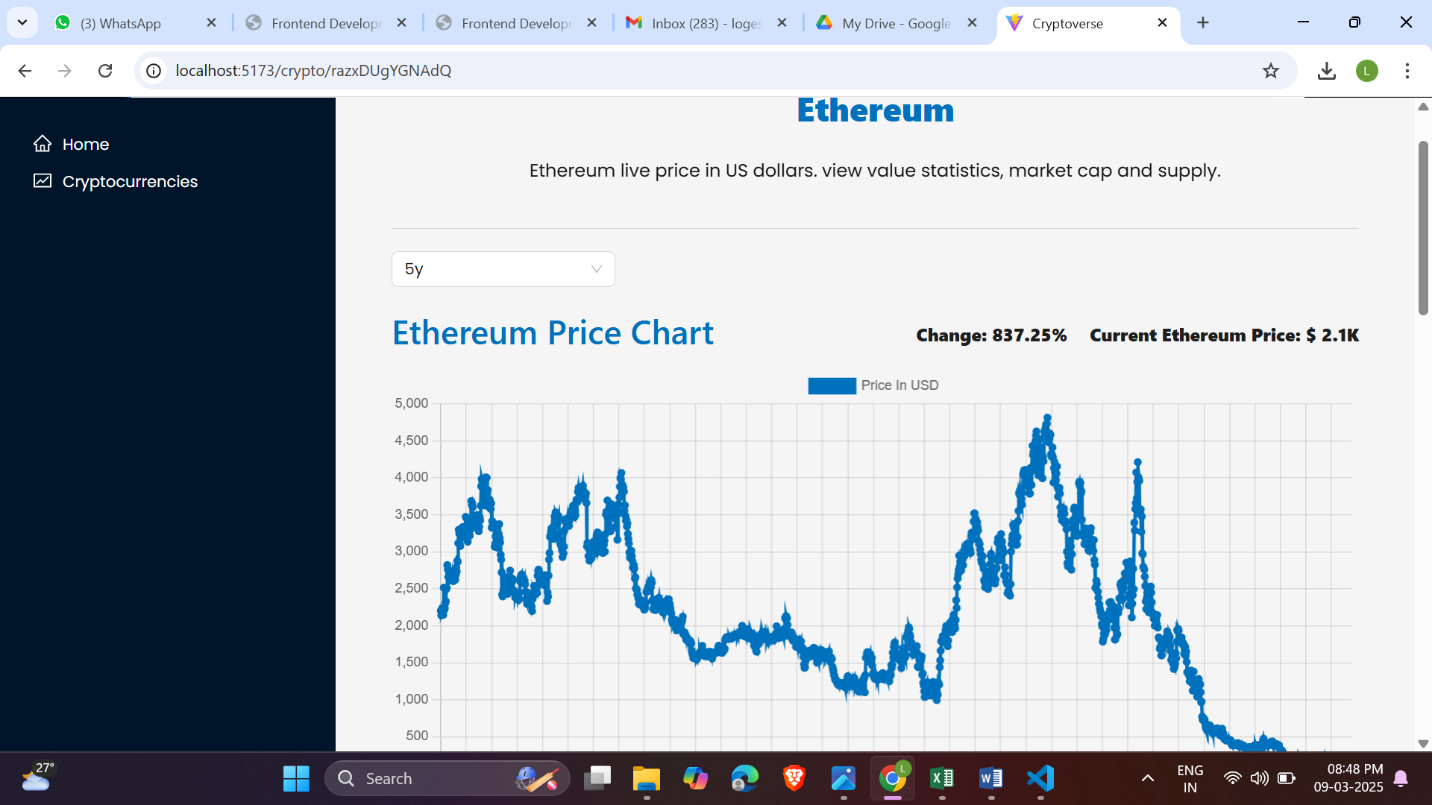
**12. Screenshots or Demo :**

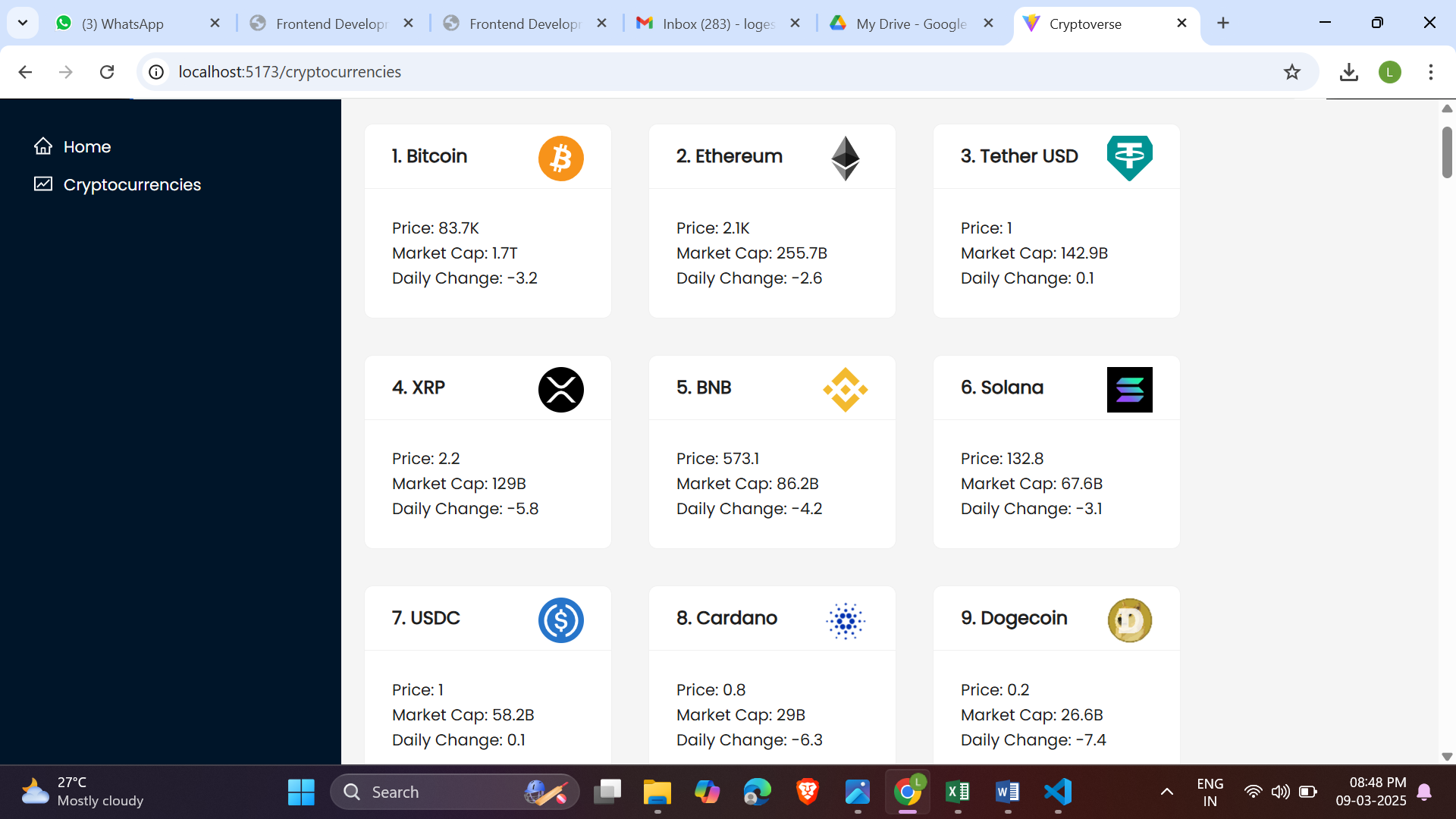
**Demo link**

[**https://drive.google.com/file/d/1Z0Mn4opeEUWeiUjkMxYVxzsv-6nfMong/view?usp=drive\_link**](https://drive.google.com/file/d/1Z0Mn4opeEUWeiUjkMxYVxzsv-6nfMong/view?usp=drive_link)

****

****

****

****

**Home Page :** Displays global crypto stats and top 10 cryptocurrencies.

**Cryptocurrencies Page :** Lists all cryptocurrencies with search functionality.

**Crypto Details Page :** Displays detailed information about a specific cryptocurrency, including historical price data and statistics.

**13. Known Issues :**

**Performance:** The application may experience performance issues when loading large datasets.

**Responsiveness:** Some components may not be fully responsive on smaller screens.

**14. Future Enhancements :**

**Enhanced Search:** Improve search functionality with filters and sorting options.

**User Authentication:** Add user authentication to allow personalized dashboards.

**Advanced Analytics:** Include more advanced analytics tools for deeper market analysis.

**Mobile Optimization:** Improve the mobile user experience with responsive design enhancements.

---

This documentation provides a comprehensive overview of the Cryptoverse cryptocurrency dashboard, covering its purpose, architecture, setup instructions, component structure, state management, user interface, and future enhancements. The application is designed to be a powerful tool for cryptocurrency investors, offering detailed historical data and insightful analysis to aid in decision-making.

|  |  |
| --- | --- |
|  |  |