

Frontend Development with React.js

Project Documentation for Cryptoverse: A Cryptocurrency Dashboard

1. Introduction

Project Title: A Cryptocurrency Dashboard

Team Members:

Team Leader : SIVARAJ V - Email Id: vsivarajcs22.krmmc@gmail.com

Team member : SEMBULINGAM A - Email Id: asembulingamcs22.krmmc@gmail.com

Team member : SURYA M - Email Id: suryams1403@gmail.com

Team member : SRINATH P - Email Id: psrinathcs22.krmmc@gmail.com

Team member : SAI KRUPESH B - Email Id: bsaikrupeshcs22.krmmc@gmail.com

2. Project Overview

Purpose

Cryptocurrency aims to provide a decentralized, secure, and efficient alternative to traditional currencies. It enables peer-to-peer transactions without intermediaries, reduces transaction fees, and offers financial inclusion to the unbanked. Cryptos also serve as a store of value, investment, and a foundation for innovations like blockchain and decentralized applications.

Features

- **Decentralized:** No central control like banks. Responsive and user-friendly UI.
- **Secure:** Uses encryption to protect transactions.
- **Private:** Some cryptos allow anonymous transactions.
- **Global:** Can be used anywhere with internet access.

3. Architecture

Component Structure

- /src
 - |— /components
 - |— Navbar.jsx # Navigation bar with links
 - |— Footer.jsx # Footer section
 - |— Sidebar.jsx # Sidebar navigation for docs
 - |— DocSection.jsx # Component for individual doc sections
 - |— MarkdownRenderer.jsx # If using markdown docs
 - |— ...

- |— /pages
- | |— Home.jsx # Home page
- | |— Details.jsx # Details page
- | |— Documentation.jsx # Main Documentation page
- | |— NotFound.jsx # 404 Page
- | |— ...

State Management

- **Context API** is used for global state management, storing coins data, user preferences, and exchange details.

Routing

- **React Router** is used to handle navigation between pages:
 - / Home Page
 - /Cryptocurrencies/:categoryName - Dynamic route for news categories

4. Setup Instructions

Prerequisites

Ensure you have the following installed:

- Node.js (latest LTS version recommended)
- npm (v8 or higher)

Installation

- Clone the repository:
- Git clone <https://github.com/sivaraj-4s/A-Cryptocurrency-Dashboard.git>
- Navigate into the project directory:
- Extract the File
- Open VS code, open your folder
- npm install:
- npm run dev

5. Folder Structure

/crypto

|— /node_modules # Dependencies installed via npm or yarn

|— /public # Static assets (not uploaded, but typically exists)

```

|   |—— vite.svg
|   |—— ...
|—— /src          # Source code for the application
|   |—— main.jsx   # Entry point referenced in index.html
|   |—— /components # Reusable React components (likely)
|   |—— /pages     # Page components (likely)
|   |—— /redux     # Redux store setup (based on dependencies)
|   |—— /utils     # Utility functions (if needed)
|   |—— /assets    # Images, styles, etc. (if needed)
|   |—— ...
|—— index.html    # Main HTML file
|—— package.json  # Project metadata and dependencies
|—— package-lock.json # Locked dependency versions
|—— README.md     # Project documentation
|—— vite.config.js # Vite configuration file
|—— .gitignore    # Git ignored files (if applicable)

```

6. Running the Application

To start the frontend server locally, run:

```
npm run dev
```

This will launch the application at <http://localhost:5173/>

7. Component Documentation

Key Components

- **index.html** → Loads the React app in `<div id="root"></div>`.
- **src/main.jsx** → Renders the app and wraps it with providers.

Reusable Components

- **src/components/** → Reusable UI components (e.g., buttons, charts).
- **src/pages/** → Main pages for routing (e.g., Home, Details)

8. State Management

Global State

- Uses **Redux Toolkit** (**@reduxjs/toolkit**) for global state.

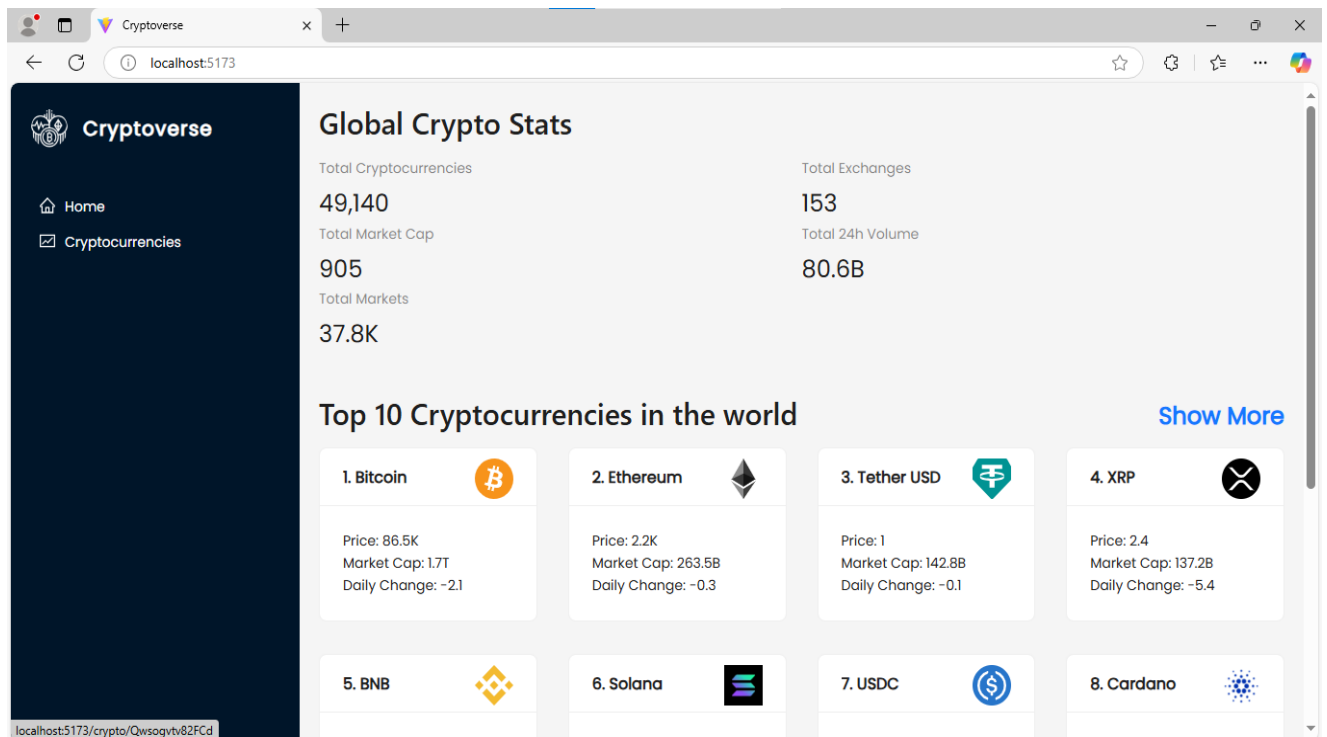
Local State

- Uses **Ant Design** (**antd**) for prebuilt UI components.

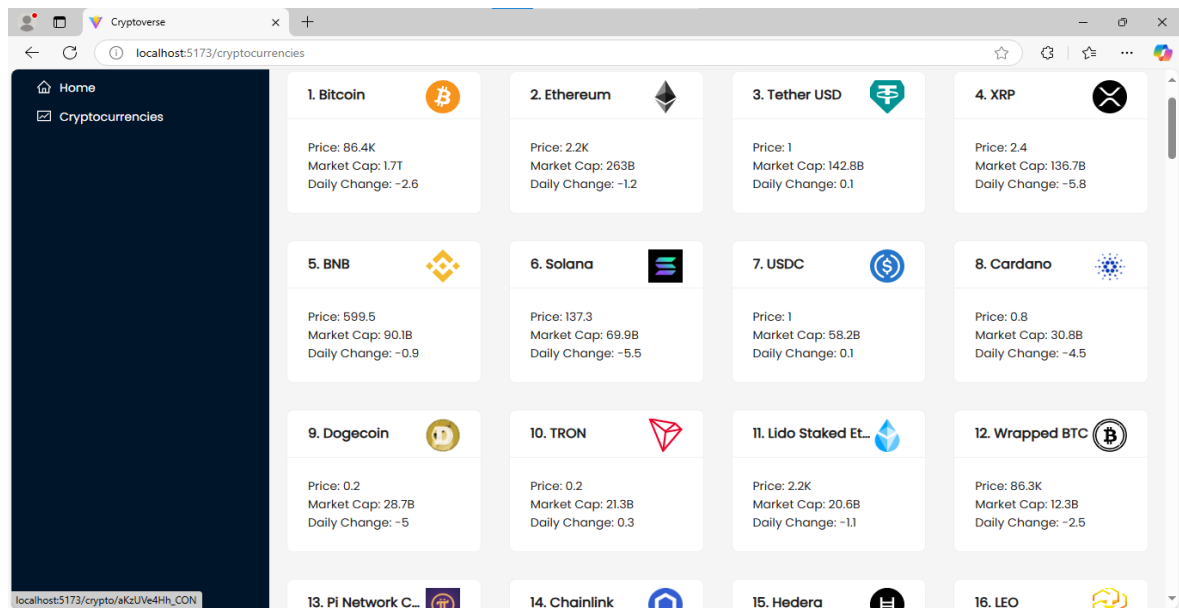
9. User Interface

Screenshots or GIFs showcasing:

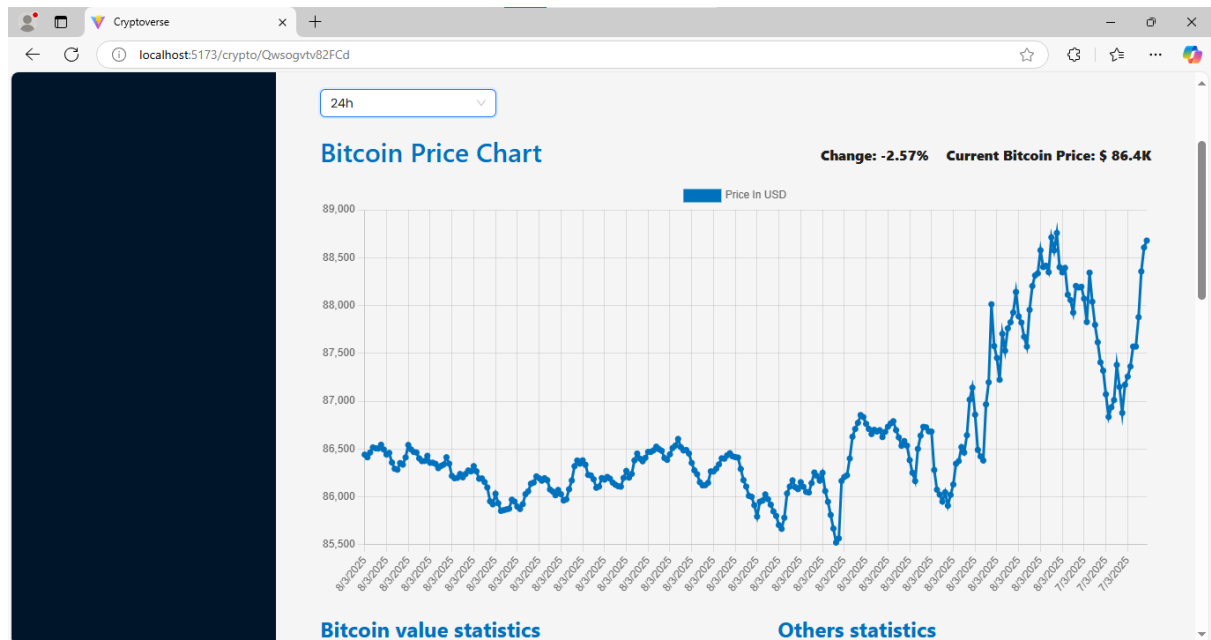
Homepage of Cryptocurrency Dashboard



Cryptocurrencies section



After Click a Bitcoin menu:



10. Styling

CSS Frameworks/Libraries

- **Styled Components** for modular and dynamic styling.
- **CSS Modules** for scoped styles in individual components.

Theming

- No theming were include.

11. Testing

Testing Strategy

- **NodeTesting Library** for unit and integration tests.
- (future enhancement) for end-to-end testing.

Code Coverage

- Code coverage reports generated using node.

12. Demo

<https://drive.google.com/file/d/1wDveVNdXd95hl8FoC5snVmUuEsbB69sD/view?usp=drivesdk>

13. Known Issues

- Performance optimizations needed for large crypto datasets.
- Improve mobile responsiveness for small-screen devices.

14. Future Enhancements

- Add a search functionality for coins.
- Improve animation and transitions for a smoother user experience.

CONCLUSION

This documentation provides a comprehensive overview of the Cryptocurrency Dashboard project, including its architecture, setup instructions, and future plans.