**Cryptoverse Documentation**

**1. Introduction**

**Project Title:**  Cryptoverse

**Team ID :** SWTID1741253118152108

**Team Members & Gmail Id:**

**NAME GMAIL ID**

Team Leader - Benzy Mabell **S** - benzymabell@gmail.com

Team Member - Hemalatha P - hemalatha7434@gmail.com

Team Member - Sajitha M - sajitha2416@gmail.com

Team Member - Zanofar Fathima A - zanosiddique491@gmail.com

**2. Project Overview**

**Purpose:**

Cryptoverse is a valuable cryptocurrency dashboard that displays historical price data over the past five years, providing investors with a comprehensive view of market dynamics. It features detailed charts and graphs for analyzing various cryptocurrencies, with customizable timeframes for in-depth price movement examination. This data aids in making informed decisions and identifying market trends, while also serving as an educational resource to enhance users' understanding of the evolving cryptocurrency landscape.

**Key Features:**

* Seamless Navigation
* Comprehensive Cryptocurrency Listings
* Visual Currency Browsing
* Interactive Charts
* Price Fluctuation Visualization
* Search Feature
* Insights and Decision-Making
* Educational Resource

**3. Architecture**

**Component Structure:**

App (Root Component)

├── Header

│ └── Navigation Links

├── Home (Main Content)

│ ├── CryptoCurrencies

│ ├── Search

│ └── Market Overview

├── CryptoDetail

│ └── Historical Data Chart

└── NotFound

**State Management:**

* Uses React's built-in useState and useEffect hooks.
* Implements axios for API calls to fetch cryptocurrency data.
* Local storage for user preferences.
* Context API or Redux for global state management.
* Real-time updates for price changes and user interactions.

**Routing:**

Uses React Router v6 with the following routes:

* / - Home page displaying an overview of the application.
* /cryptocurrencies - List of available cryptocurrencies.
* /cryptocurrency/:id - Detailed view of a specific cryptocurrency.
* \* - NotFound page for undefined routes.

**4. Setup Instructions**

**Prerequisites**

- Node.js (v14 or higher)

- npm (v6 or higher)

- JSON Server

**Installation**

**Install dependencies**

cd cryptoverse

npm install

npm install react

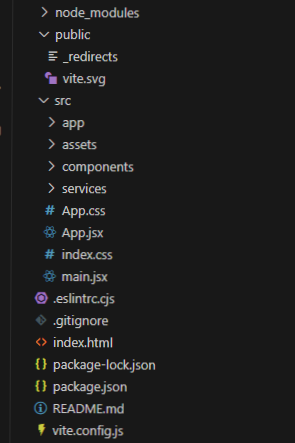
**Start JSON Server**

json-server --watch db.json --port 3000

**Start the application**

npm run dev

**5. Folder Structure**

****

**6. Running the Application**

**Start the development server**

npm run dev

**Start the JSON server**

npx json-server --watch db.json

**7. Component Documentation**

* **node\_modules**: Contains all the installed dependencies for project.
* **public:**
  + \_redirects: Typically holds redirect rules for web hosting.
  + vite.svg: An SVG image file, possibly the logo or related to Vite (a build tool).
  + **src (Source Files):**
* app: Likely contains the main application logic.
* assets: Stores images, fonts, and other static files.
* components: Contains reusable UI components.
* services: Could include modules for handling data, API calls, etc.

**Common Files in src:**

* App.css: Styles for the main application.
* App.jsx: Main component for the React application.
* index.css: Global styles.
* main.jsx: Entry point of the React application.

**Configuration and Metadata Files:**

* .eslintrc.cjs: Configuration file for ESLint, a tool to find and fix problems in JavaScript code.
* .gitignore: Specifies files and directories that should be ignored by Git.
* index.html: The main HTML file that loads the application.
* package-lock.json: Locks the versions of installed packages.
* package.json: Contains metadata about the project and lists dependencies.
* README.md: Documentation for the project.
* vite.config.js: Configuration file for Vite.

**8. State Management**

**Global State:**

* User authentication
* Cryptocurrency data
* Global settings
* Transaction history

**Local State:**

* Form inputs
* UI toggle states
* Component-specific details
* Temporary data

**9. User Interface**

* Blends dark and light themes for improved readability.
* Clear sections for quick stats and detailed information.
* User-friendly navigation with a menu for easy access to other sections.

**10. Styling**

* CSS Framework
* Bootstrap 5.3.2
* Custom CSS modules
* Responsive design principles

**11. Testing**

Testing Strategy

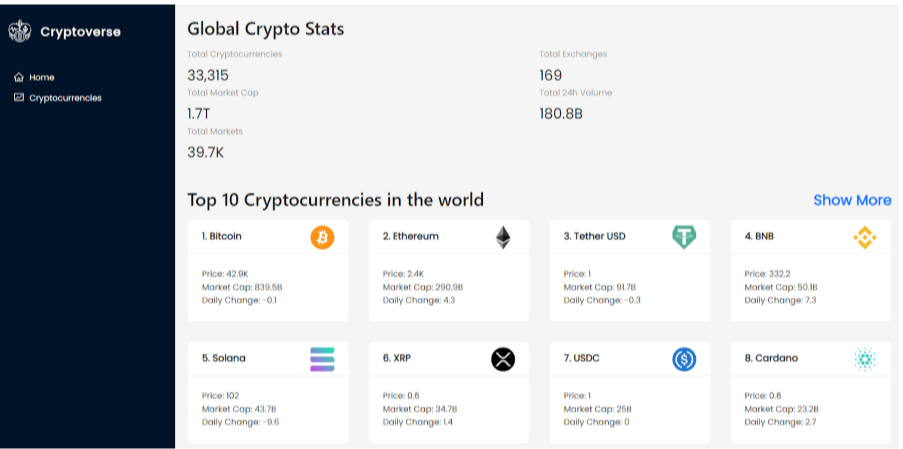
* Component testing using React Testing Library
* Integration tests for API calls
* User interaction testing

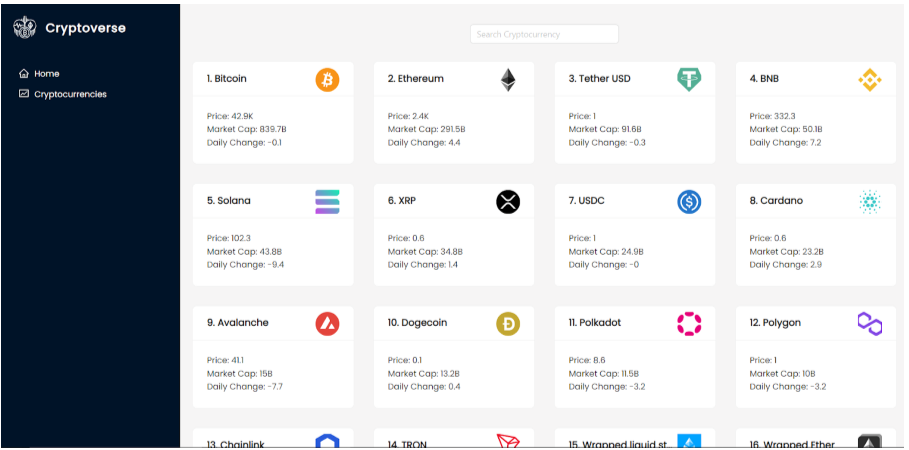
Test Coverage

Component rendering

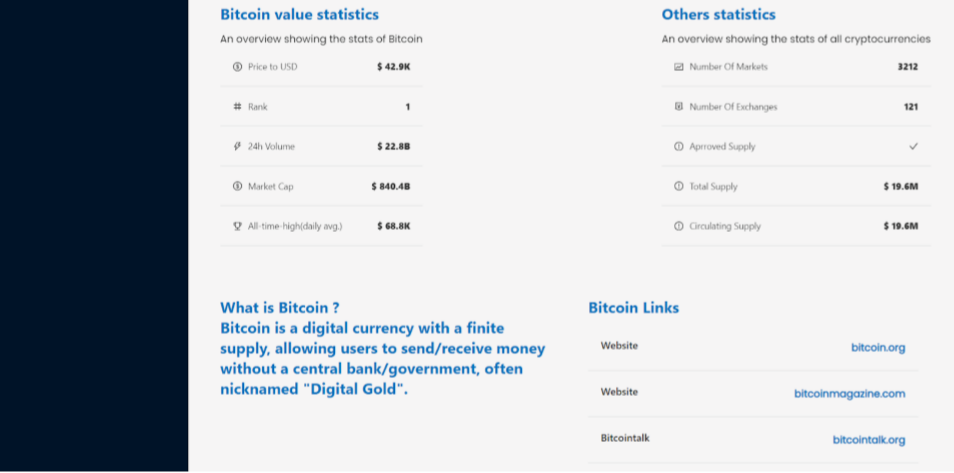
* User interactions
* State management
* API integration

**12. Screenshots**









**13. Known Issues**

* User Interface (UI) Concerns
* Data Accuracy
* Performance Issues
* Accessibility Features
* Error Handling
* Mobile Optimization
* Documentation and Support

**14. Future Enhancements**

**Community Features:**

Include sections for community discussions, news, and updates related to specific cryptocurrencies.

Implement integration with social media platforms for sharing insights.

**Educational Resources:**

Offer guides, tutorials, and articles for beginners to learn about cryptocurrency investing.

Develop a glossary of terms commonly used in the cryptocurrency market.

**Portfolio Management:**

Introduce features that allow users to track their cryptocurrency investments and portfolios directly within the dashboard.

**Integration with Wallets:**

Enable users to connect their cryptocurrency wallets for real-time balance tracking and transaction history.

**Regulatory Updates:**

Incorporate a section dedicated to global regulatory updates affecting the cryptocurrency market.

**Links**

**GitHub :**

https://github.com/BenzyMabell/Cryptoverse

**Google drive link:**

https://drive.google.com/drive/folders/1jDF\_ZWhWGVSFmLSwvC2-PF2gy7QVMpyq?usp=sharing

**Demo video link:**

https://drive.google.com/file/d/1V7mDG4Zsbw3jt9AqSuTaXWA0kePUGmqe/view?usp=sharing