CRYTOVERSE(CRYTOCURRENCY)

Project Documentation

1. **Introduction:**

A cryptocurrency dashboard with five years of data enables trend analysis, comparisons, and informed decisions through intuitive charts and customizable timeframes. It also serves as an educational tool.

Project Title:

**Crytoverse-Crytocurrency**

**Team Members:**

**Team ID : SWTID1741235636151174**

Team Size :4

Team Leader : POOJASRI S ([poojasri20005@gmail.com](mailto:poojasri20005@gmail.com))

Team member : PRIYADHARSHINI V ([sd8471108@gmail.com](mailto:sd8471108@gmail.com) )

Team member : DEEPIKA I ([deepikasaraswathi800@gmail.com](mailto:deepikasaraswathi800@gmail.com) )

Team member : RAHIMUNISA BEGUM J ([rahimunissabgm786@gmail.com](mailto:rahimunissabgm786@gmail.com) )

**2. Project Overview**

Purpose:

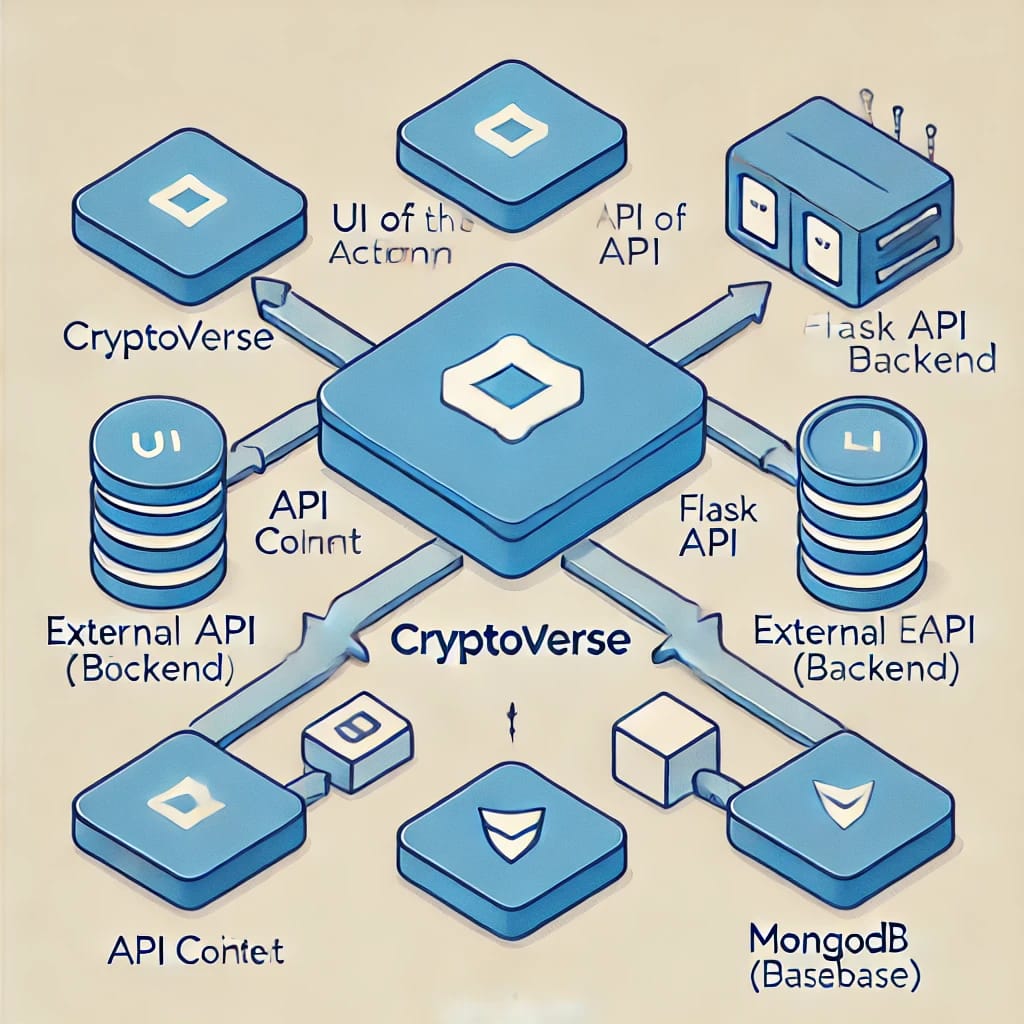
cryptocurrency dashboard designed to provide investors with comprehensive insights into market dynamics through detailed historical price data analysis spanning five years.

Features:

* Historical Price Data Analysis
* Interactive Charts & Graphs
* Comparison of Multiple
* Customizable Time Frames
* Risk Assessment & Pattern
* Robust Search Functionality

**3. Architecture**

­­­­­



Component Structure:

* The Frontend has Reusable components such as Navbar,Dashboard,Converter,Search and Authentication.
* The Backend Manages API Services,Data Processing and Database Interactions.

State Management:

* Uses Local State for UI Interactions and Global State for User Data and Settings.

Routing:

* Frontend : Handles Navigation
* Backend : Manages Requests for Cryto data, and conversions and User Preferences.

**4. Setup Instructions**

Prerequisites:

Node.js (latest stable version)

npm

React.js

Firebase (for authentication and database)

**Installation:**

1. Clone the repository:

git clone: https://github.com/asunm13332213331026023/cryptocurrency.git

2. Navigate to the project directory:

cd code

3. Install dependencies:

npm install

4. Configure environment variables (Firebase API keys, etc.).

5. Start the application:

npm start

**5. Folder Structure**

Client:

src/components: Reusable UI components (buttons, cards, modals)

src/pages: Individual screens (Home, Dashboard, Profile, etc.)

src/services: API calls and data fetching

src/context: Context API and state management

src/assets: Images, icons, and styles

Utilities:

Helper functions for formatting data, API integration, and authentication handling.

**6. Running the Application:**

Start the frontend server:

npm start

The app will be available at http://localhost:3000.

**7. Component Documentation**

Key Components:

* Navbar: Navigation bar for switching between pages.
* Converter : Converts Crypttcurrency to Fiat Currency based on Live Exchange rates.
* Dashboard : Displays Cryptocurrency Data(Prices,Trends,Historical Data)
* Reusable Components:

Button: Customizable Button used across the App.

InputField : A Reusable input fields for search,cryptocurrency input and forms .

Loader : Display the loading animation when fetching data.

**8. State Management**

Global State:

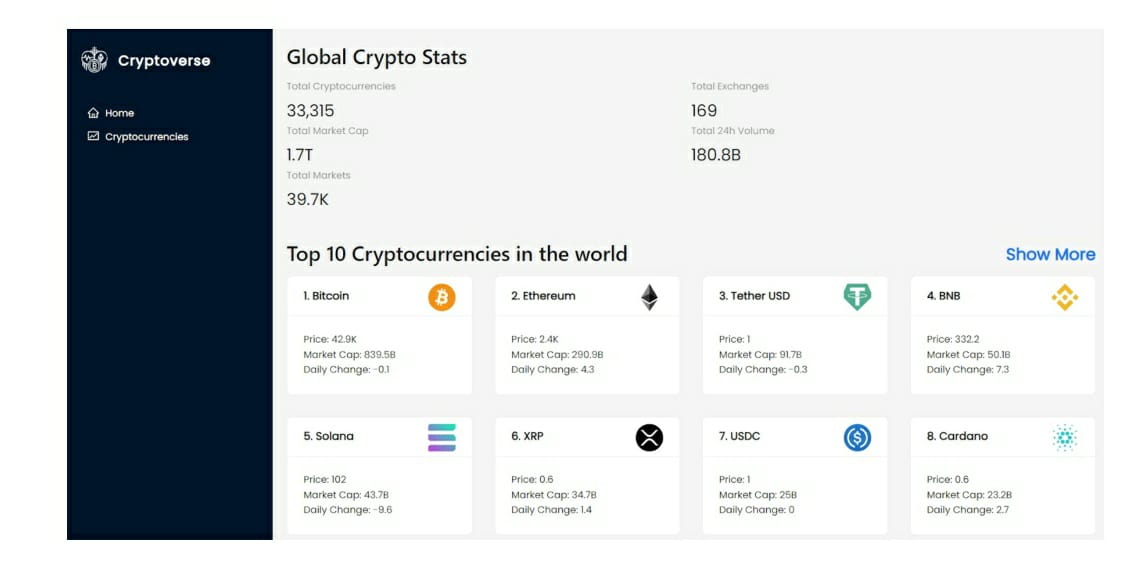
* Used for managing global data like crytocurrency prices,user preferences and API responses.

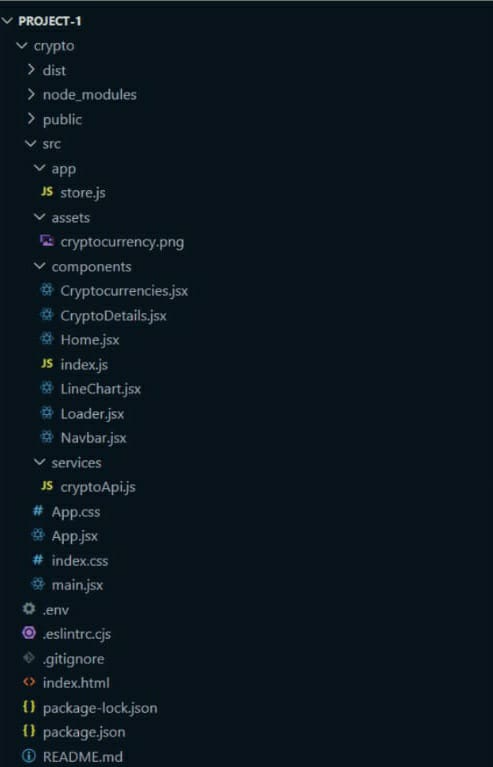
Local State:

* Used for managing UI – Related data that doesn’t need to be shared globally.

**9. User Interface**

Screenshots/GIFs:





**10. Styling**

CSS Frameworks/Libraries:

* Tailwind CSS for utility-based styling.
* Styled-Components for dynamic theming.

**Theming :**

* Dark/Light Mode Toggle – Uses CSS variables to switch themes dynamically.
* Custom Design System – Ensures consistency in typography, colors, and UI components.
* Styled Components & CSS Variables – Used for flexible and scalable theming.

**11. Testing**

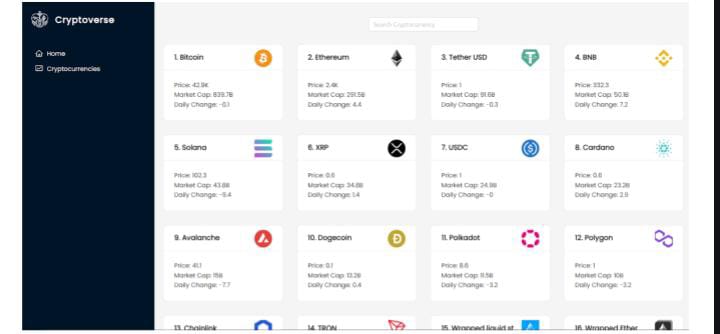
Testing Strategy:

* Unit testing with Jest and React Testing Library.
* Integration testing.
* End-to-end testing with Cypress.
* Performance and security testing.

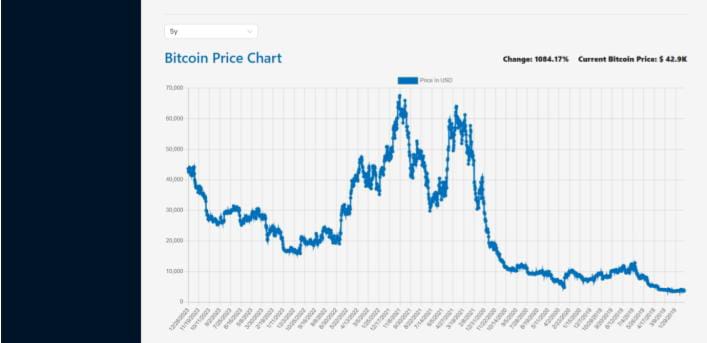
**Code Coverage:**

* Jest with -- coverage – Measures test coverage for components and functions.
* Metrics Tracked – Statements, branches, functions, and lines covered.
* Threshold Enforcement – Set minimum coverage in package.json.
* ­­­Report Formats – Generates HTML, JSON, CLI output for analysis.

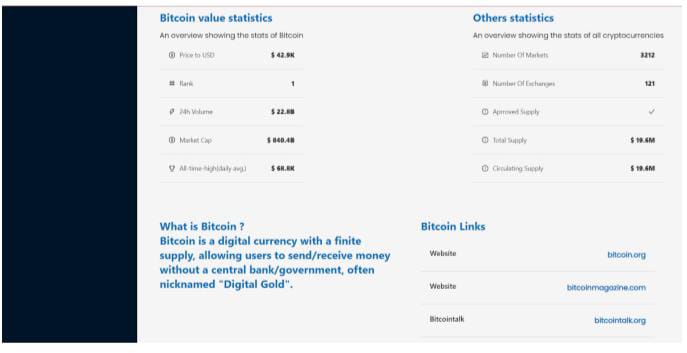
**12. Screenshots or Dem**

****

**Live Demo:**

****

­­­­­­



**13. Known Issues**

* API Rate Limits – Too many requests may cause temporary blocks.
* Slow Initial Load – Large data fetching affects performance.
* Theme Toggle Issue – Dark/Light mode may not persist on refresh.
* Mobile UI Bugs – Some elements may not scale properly.
* Redux State Reset – Loses state on reload without localStorage.

**14. Future Enhancements**

**Future Enhancements for Cryptoverse**

**1. Real-Time Price Updates (WebSocket Integration)**

* Issue: Prices update at intervals, causing delays.
* Solution: Use WebSockets (e.g., Binance API) for instant price updates.
* Benefit: Live, real-time price changes without frequent API calls.

**2. Advanced Charting & Technical Indicators**

* Issue: Charts display basic trends but lack detailed analysis.
* Solution: Integrate TradingView API or enhance Chart.js/D3.js with indicators like RSI, MACD, Bollinger Bands.
* Benefit: Helps traders with in-depth market analysis.

**3. Dark/Light Mode Persistence**

* Issue: Theme resets on page refresh.
* Solution: Store user preferences in localStorage or use Redux Persist.
* Benefit: Saves user settings for a better experience.

**4. Improved Mobile UI & Responsiveness**

* Issue: Some UI elements break on smaller screens.
* Solution: Optimize layout using CSS Grid, Flexbox, and media queries.
* Benefit: Enhanced mobile usability and performance.

**5. Portfolio Tracker & Watchlist**

* Issue: Users can view prices but cannot track holdings.
* Solution: Allow users to add cryptocurrencies to a portfolio and monitor selected assets.
* Benefit: Personalized portfolio tracking.

**Social challenges and leaderboards.**

**1. Crypto Trading Challenges**

Concept: Users participate in simulated trading competitions, trying to achieve the highest portfolio growth within a specific period (e.g., daily, weekly, monthly).

**How It Works:**

* Users get a virtual balance (e.g., $100,000) to trade real-time crypto prices.
* They make buy/sell decisions based on market conditions.
* A dynamic leaderboard updates rankings based on profit/loss.
* The top performers at the end of each challenge win badges or rewards.

**Benefits:**

* Users get a virtual balance (e.g., $100,000) to trade real-time crypto prices.
* They make buy/sell decisions based on market conditions.
* A dynamic leaderboard updates rankings based on profit/loss.
* The top performers at the end of each challenge win badges or rewards.

**2. Leaderboards & Rankings**

* Concept: A public leaderboard ranks users based on performance metrics such as profit, trade volume, and accuracy of predictions.

**Leaderboard Metrics:**

* Highest Portfolio Growth – Users with the best % increase in portfolio value.
* Best Trader – Most successful trades based on profit/loss ratio.
* Most Active Trader – Highest number of trades executed.
* Best Predictions – Most accurate price forecasts.

**How It Works:**

* Leaderboard updates in real time.
* Users can filter rankings by timeframe (daily, weekly, all-time).
* Top-ranked users earn badges, ranks, and profile recognition.

**Benefits:**

* Encourages healthy competition.
* Helps users track their performance over time.
* Recognizes and rewards top performers.

**3. Social Sharing & Community Challenges**

* Concept: Users can invite friends to join trading challenges and share their rankings on social media.

**How It Works:**

* Users create custom challenges and invite friends.
* Each challenge has specific rules (e.g., trade only Bitcoin, highest return wins).
* Social Media Sharing – Users can share achievements on Twitter, Reddit, and Discord.
* Public & Private Challenges – Compete with friends or the global community

**Benefits:**

* Encourages word-of-mouth promotion.
* Builds a community of traders who learn together.
* Attracts new users through viral challenges.

**4. Rewards & Achievement Badges**

* Concept: Users earn badges and rewards for completing challenges and reaching milestones.

**Example Badges & Rewards:**

* "Crypto Master" – Awarded to the top trader of the month.
* "HODL King" – Earned by holding a crypto asset for the longest time.
* "High-Risk Trader" – Given to users who make high-risk, high-reward trades.
* "Market Analyst" – Awarded for accurate price predictions.

**How It Works:**

* Users earn XP points, badges, or ranks based on activity.
* Badges appear on leaderboards and user profiles.
* Potential rewards include premium features, tokens, or exclusive content.

**Benefits:**

* Keeps users motivated to participate.
* Recognizes top-performing traders.
* Encourages users to improve their skills over time.

**5. Referral System & Invite Rewards**

* Concept: Users earn bonuses by referring friends to join Cryptoverse.

How It Works:

* Each user gets a unique referral link.
* When a friend joins and participates, the referrer gets bonus XP or rewards.
* Special referral leaderboards track top referrers.

**Benefits:**

* Helps grow the user base organically.
* Encourages community-driven adoption.
* Rewards loyal users for bringing in new members.

**6. Technical Implementation**

* Frontend: React-based UI with leaderboard components and dynamic ranking updates.
* Backend: Flask-based API storing user stats, challenge data, and referral info in MongoDB.
* Database: MongoDB stores user rankings, trade history, and achievements.
* WebSockets (for Real-Time Updates): Used to update leaderboards dynamically.