# PROJECT NAME

# CRYPTOVERSE DASHBOARD


## TEAM LEADER:

JAYANTHI.S ( jayanthijaya347@gmail.com)

## TEAM MEMEBERS

SHALINI.D (Shaliniblak@gmail.com)

PONNARUVI.R (ponnaruviraja@gmail.com)

VELVIZHI.E (velvizhiswetha@gmail.com)

# SUBMIT DATA: 08/03/2025

## INTRODUCTION:

A crypto currency dashboard that displays historical price data over the past five years is a powerful tool for investors seeking a comprehensive understanding of market dynamics. This feature-rich interface offers users a detailed historical perspective on the performance of various crypto currencies, enabling insightful analysis and informed decision-making. Through visually intuitive charts and graphs, the dashboard allows for effective comparisons of multiple crypto currencies, aiding in the identification of top performers and overall market trends. Users can customize timeframes for a more granular examination of price movements, facilitating in-depth volatility analysis and risk assessment.

This historical data not only supports investors in making data-driven decisions but also assists in recognizing recurring patterns and cycles. Beyond its role in optimizing cryptocurrency portfolios, the dashboard serves as an educational resource, empowering users to grasp the evolving nature of crypto currency markets and the nuanced factors shaping price movements over an extended period.

The Cryptoverse Dashboard is an all-in-one platform for navigating the world of cryptocurrency and blockchain. Our dashboard provides real-time data, analytics, and insights to help you make informed decisions in the cryptoverse.

## Real-time Market Data:

Get up-to-the-minute prices, trading volumes, and market capitalizations for thousands of cryptocurrencies.

## Portfolio Tracking:

Easily track your cryptocurrency holdings and performance across multiple exchanges and wallets.

## Blockchain Analytics:

Dive deep into blockchain data, including transaction volumes, block times, and network congestion.

## News and Insights:

Stay ahead of the curve with our curated feed of cryptocurrency news, research, and analysis.

## Customizable Dashboards:

Tailor your dashboard to your needs, with customizable widgets, layouts, and alerts.

# OBJECTIVES:

## Provide Real-time Market Data:

Offer accurate, up-to-date, and comprehensive market data to empower informed decision-making.

## Develop Advanced Analytics:

Create sophisticated tools and metrics for analyzing market trends, trading activity, and portfolio performance.

## Ensure Transparency and Security:

Implement robust security measures and provide transparent information to foster trust and confidence.

## Simplify User Experience:

Design an intuitive, user-friendly interface that streamlines navigation and facilitates engagement.

## Foster Community Engagement:

Create a platform for users to connect, share knowledge, and collaborate.

# PROBLEM STATEMENT:

The cryptocurrency market is highly volatile, decentralized, and rapidly evolving, making it challenging for investors, traders, and enthusiasts to navigate and make informed decisions. Current solutions often provide fragmented, outdated, or biased information, leading to missed opportunities, financial losses, and a lack of confidence in the market.

### Information Overload:

The sheer volume of data from various sources makes it difficult to identify relevant and reliable information.

### Lack of Transparency:

Insufficient visibility into market trends, trading activity, and project developments hinders informed decision-making.

### Limited Analytics:

Inadequate tools and metrics for analyzing market data and portfolio performance restrict users' ability to optimize their investments.

**Security Concerns:** The risk of hacking, phishing, and other security threats discourages users from engaging with the cryptocurrency market.

5. User Experience: Cluttered, complex, and poorly designed interfaces deter users from exploring the cryptocurrency market.

# types of cryptocurrencies:

# Bitcoin:

Bitcoin is a decentralized cryptocurrency that operates without a central authority or single administrator. It was invented in 2008 by an unknown entity under the pseudonym Satoshi Nakamoto and began operating in 2009.

## Key Characteristics:

- Decentralized: Bitcoin operates on a peer-to-peer network, with no central authority controlling it.

- Limited Supply: The total supply of bitcoin is capped at 21 million.

- Fast and Global: Bitcoin transactions are processed and settled in real-time, regardless of the sender's and recipient's locations.

- Secure: Bitcoin transactions are secured through a process called "mining," which uses advanced cryptography to verify and record transactions.

- Pseudonymous: Bitcoin transactions are pseudonymous, meaning that users can make transactions without revealing their real-world identities.

## How it Works:

1. Mining: New bitcoins are created through a process called "mining," in which specialized computers solve complex mathematical problems.

2. Transactions: Bitcoin transactions are made by sending bitcoins from one address to another.

3. Blockchain: All bitcoin transactions are recorded on a public ledger called the blockchain.

4. Wallets: Bitcoin users can store their bitcoins in digital wallets, which can be accessed through software or hardware.

## Current Status:

As of March 6, 2025, the closing price of bitcoin was $89,931.89.[2] Bitcoin has become increasingly popular as a form of investment and has been recognized as a form of currency by some governments. However, its value can be highly volatile, and its use as a form of payment is still limited compared to traditional currencies..

# Ethereum

Ethereum is an open-source, decentralized, blockchain-based platform that enables the creation of smart contracts and decentralized applications (dApps). It was founded in 2014 by Vitalik Buterin and has since become one of the largest and most widely-used blockchain platforms.

## Key Characteristics

1. Decentralized: Ethereum operates on a decentralized network of nodes, with no central authority controlling it.

2. Smart Contracts: Ethereum enables the creation of smart contracts, which are self-executing contracts with the terms of the agreement written directly into code.

3. Decentralized Applications (dApps): Ethereum enables the creation of dApps, which are applications that run on the blockchain and are decentralized.

4. Ether (ETH): Ether is the native cryptocurrency of the Ethereum platform and is used to pay for transaction fees and computational services.

## How it Works

1. Blockchain: Ethereum's blockchain is a decentralized, distributed ledger that records all transactions and smart contract executions.

2. Mining: Ethereum uses a proof-of-work (PoW) consensus algorithm, which requires miners to solve complex mathematical problems to validate transactions and create new blocks.

3. Smart Contract Execution: Smart contracts are executed by the Ethereum Virtual Machine (EVM), which runs on each node of the network.

4. Transaction Fees: Transaction fees are paid in Ether and are used to incentivize miners to validate transactions.

## Current Status

As of March 6, 2025, the market capitalization of Ether is over $500 billion, making it the second-largest cryptocurrency by market capitalization [1]. Ethereum has become a widely-used platform for decentralized applications, non-fungible tokens (NFTs), and decentralized finance (DeFi) applications.

## Future Developments

1. Ethereum 2.0: Ethereum is currently undergoing a major upgrade to Ethereum 2.0, which will transition the platform from a PoW consensus algorithm to a proof-of-stake (PoS) consensus algorithm.

2. Sharding: Ethereum 2.0 will also introduce sharding, which will enable the platform to process multiple transactions in parallel, increasing its scalability.

3. Layer 2 Scaling Solutions: Ethereum is also exploring layer 2 scaling solutions, such as Optimism and Polygon, which will enable faster and cheaper transactions.

## Tether:

Tether USD (USDT) is a stablecoin pegged to the US Dollar, meaning its value is tied to the value of the US Dollar. It's the most widely used stablecoin and is available on multiple blockchains, including Ethereum, Tron, EOS, Algorand, and more.

- Market Capitalization: Over $142 billion, making it the 4th largest cryptocurrency by market capitalization.

- Circulating Supply: Approximately 142.8 billion USDT tokens are in circulation.

- Trading Volume: The 24-hour trading volume of USDT is over $90 billion, indicating a highly liquid market.

- Price Stability: USDT is designed to maintain a 1:1 peg with the US Dollar, ensuring price stability.

- Use Cases: USDT is widely used for trading, arbitrage, and as a hedge against market volatility. It's also used for cross-border payments and remittances.

Tether USD is controlled by Tether Holding Limited, and its reserves are backed by actual fiat currencies. The company publishes its balances and reserves on its transparency page.

# Salana:

Solana (SOL) is a highly functional open-source project that utilizes blockchain technology to provide decentralized finance (DeFi) solutions. It's designed to facilitate the creation of decentralized applications (dApps) and boasts fast speeds and affordable costs.

## Key Statistics:

- Current Price: $141.23, with a 5.62% increase in the last 24 hours.

- Market Capitalization: $71.84 billion, ranking #6 in the cryptocurrency market.

- Circulating Supply: 508.7 million SOL tokens.

- Total Supply: 595.58 million SOL tokens.

## What Makes Solana Unique?

Solana combines the Proof-of-History (PoH) consensus mechanism with the Proof-of-Stake (PoS) consensus, enabling fast and secure transactions. This hybrid consensus model allows Solana to solve the blockchain trilemma, balancing decentralization, security, and scalability.

## Solana supports various use cases, including:

- Decentralized Applications (dApps): Solana's platform enables the creation of fast and secure dApps.

- Non-Fungible Tokens (NFTs): Solana's platform supports the creation and trading of NFTs.

Overall, Solana has established itself as a prominent player in the cryptocurrency market, offering a unique blend of speed, security, and scalability.

# XRP:

XRP, also known as Ripple, is a decentralized cryptocurrency that operates on the XRP Ledger (XRPL) blockchain. It's designed for fast, low-cost, and scalable transactions, making it ideal for cross-border payments and financial institutions.

## Key Statistics:

- Current Price: $2.52, with a 3.33% increase in the last 24 hours.

- Market Capitalization: Over $146 billion, ranking it among the top cryptocurrencies.

- Circulating Supply: Approximately 57.94 billion XRP tokens.

- Total Supply: 100 billion XRP tokens

## How XRP Works:

XRP facilitates transactions on the XRPL network, protects the ledger from spam, and bridges currencies in the XRP Ledger's native decentralized exchange (DEX).[2] The XRPL uses a consensus protocol to validate transactions, which are processed in as little as 3-5 seconds.

## XRP is used for various purposes, including:

- Cross-Border Payments: Financial institutions can use XRP for fast and cheap international transactions.

- Decentralized Exchange (DEX): XRP facilitates currency exchange on the XRPL's native DEX.

- Tokenization: The XRPL enables the tokenization of assets, making it easier to create and manage digital assets.

Overall, XRP is a fast, secure, and scalable cryptocurrency that's gaining traction in the financial industry.

# USDC:

USD Coin (USDC) is a stablecoin pegged to the US Dollar, issued by Circle and Coinbase. It's designed to provide a stable and secure way to store and transfer value.

## Key Statistics

1. Current Price: $1.00, with a 0.01% change in the last 24 hours.

2. Market Capitalization: Over $55 billion, ranking it among the top stablecoins.

3. Circulating Supply: Approximately 55.6 billion USDC tokens [1]

4. Total Supply: 55.6 billion USDC tokens.

## How USDC Works

1. Collateralization: USDC is backed by a reserve of US Dollars, held in a segregated account.

2. Smart Contract: The USDC smart contract ensures that each USDC token is redeemable for one US Dollar.

3. Auditing: The USDC reserve is audited monthly by Grant Thornton, a leading accounting firm.

## Use Cases

1. Stable Store of Value: USDC provides a stable store of value, protected from market volatility.

2. Cross-Border Payments: USDC enables fast and cheap cross-border payments.

3. Decentralized Finance (DeFi): USDC is widely used in DeFi applications, such as lending and borrowing.

### Benefits

1. Stability: USDC is pegged to the US Dollar, ensuring stability and predictability.

2. Security: USDC is backed by a reserve of US Dollars and audited regularly.

3. Liquidity: USDC is widely available on cryptocurrency exchanges and DeFi platforms.

## Cardana:

Cardano (ADA) is a decentralized, open-source blockchain platform that utilizes a proof-of-stake (PoS) consensus algorithm to secure its network. It's designed to facilitate the creation of decentralized applications (dApps) and provide a scalable, secure, and sustainable platform for the development of blockchain-based solutions.

### Key Statistics

1. Current Price: $0.38, with a 2.11% increase in the last 24 hours.

2. Market Capitalization: Over $13.4 billion, ranking it among the top cryptocurrencies.

3. Circulating Supply: Approximately 34.4 billion ADA tokens.

4. Total Supply: 45 billion ADA tokens.

## How Cardano Works

1. Ouroboros Consensus Algorithm: Cardano uses a proof-of-stake (PoS) consensus algorithm called Ouroboros, which is designed to be energy-efficient and secure .

2. Decentralized Governance: Cardano has a decentralized governance model, which allows holders of ADA tokens to participate in the decision-making process .

3. Smart Contracts: Cardano supports the creation of smart contracts, which can be used to automate various processes and create decentralized applications (dApps) .

## Use Cases

1. Decentralized Applications (dApps): Cardano provides a platform for the creation of dApps, which can be used in various industries such as finance, healthcare, and supply chain management .

2. Decentralized Finance (DeFi): Cardano supports the creation of DeFi applications, which can provide financial services such as lending, borrowing, and trading .

3. Identity Verification: Cardano's blockchain can be used for identity verification, providing a secure and decentralized way to manage identity documents .

### Future Developments

1. Vasil Hard Fork: Cardano is planning to implement a hard fork called Vasil, which will bring significant improvements to the network's scalability and performance .

2. Layer 2 Scaling Solutions: Cardano is exploring layer 2 scaling solutions, which will enable faster and cheaper transactions on the network .

3. Decentralized Governance: Cardano is continuing to develop its decentralized governance model, which will enable holders of ADA tokens to participate in the decision-making process.

## Avalanche:

Avalanche (AVAX) is a decentralized, open-source blockchain platform that facilitates the creation of decentralized applications (dApps) and custom blockchain networks. It's designed to be highly scalable, with transaction finality times of under one second.

## Key Statistics:

- Current Price: $20.61, with a 7.83% increase in the last 24 hours.

- Market Capitalization: $8.53 billion.

- Circulating Supply: 414.18 million AVAX tokens.

- Total Supply: 450.85 million AVAX tokens.

- Consensus Algorithm: Proof-of-Stake (PoS).

## Use Cases:

- Decentralized Applications (dApps): Avalanche provides a platform for creating custom blockchain networks and dApps.

- Decentralized Finance (DeFi): Avalanche supports DeFi applications, including lending, borrowing, and trading.

- Custom Blockchain Networks: Avalanche allows users to create custom blockchain networks tailored to their specific needs.

## Unique Features:

- High Scalability: Avalanche's novel architecture enables high scalability without sacrificing speed, reliability, or security.

- Low Energy Consumption: Avalanche consumes the same energy as only 46 US households each year, making it an eco-friendly option.

- Interoperability: Avalanche supports interoperability with other blockchain platforms, including Ethereum.

# Dogecoin:

Dogecoin (DOGE) is a decentralized, open-source cryptocurrency that was created as a joke in December 2013 by software engineers Billy Markus and Jackson Palmer.[1] [2] It's based on the popular "Doge" internet meme featuring a Shiba Inu dog. Despite its satirical nature, Dogecoin has gained a large following and is considered a legitimate investment prospect.

## Key Statistics:

- Current Price: $0.2066

- Market Capitalization: Over $30.65 billion

- Circulating Supply: Approximately 148.32 billion DOGE tokens

- Total Supply: Unlimited, with exactly 5 billion new tokens entering circulation each year

## Use Cases:

- Online Tipping: Dogecoin is popular for online tipping on platforms like Reddit and Twitter.

- E-commerce: Some merchants accept Dogecoin as a form of payment.

- Charity: The Dogecoin community has raised funds for various charities, including the Jamaican Bobsled Team and Doge4Water.

### Notable Features:

- Fast Transaction Times: Dogecoin's block time is 1 minute, making it faster than Bitcoin's 10-minute block time.

- Low Fees: Transaction fees on the Dogecoin network are relatively low.

- Active Community: Dogecoin has a dedicated and active community, with many enthusiasts and supporters.

## Palkadal:

1. It's a new or emerging cryptocurrency: If Palkadal is a new cryptocurrency, it may not have gained enough traction or attention to be listed on popular cryptocurrency platforms or websites.

2. It's a misspelling or variation: Please double-check the spelling or name of the cryptocurrency you're interested in.

3. It's not a publicly traded cryptocurrency: Palkadal might be a private or internal cryptocurrency, not publicly traded or listed.

# Polygon:

Polygon (MATIC) is a decentralized, open-source blockchain platform that utilizes a proof-of-stake (PoS) consensus algorithm to secure its network. It's designed to facilitate the creation of decentralized applications (dApps) and provide a scalable, secure, and sustainable platform for the development of blockchain-based solutions.

## Key Statistics:

- Current Price: $0.2482, with a -0.94% change in the last 24 hours

- Market Capitalization: $474.91 million

- Circulating Supply: Approximately 1.91 billion MATIC tokens

- Total Supply: 10 billion MATIC tokens

## Use Cases:

- Decentralized Applications (dApps): Polygon provides a platform for creating custom blockchain networks and dApps.

- Decentralized Finance (DeFi): Polygon supports DeFi applications, including lending, borrowing, and trading.

- Custom Blockchain Networks: Polygon allows users to create custom blockchain networks tailored to their specific needs.

### Unique Features:

- Scalability: Polygon's novel architecture enables high scalability without sacrificing speed, reliability, or security.

- Proof-of-Stake Consensus Algorithm: Polygon uses a PoS consensus algorithm, which is more energy-efficient compared to traditional proof-of-work (PoW) algorithms.

- Ethereum Compatibility: Polygon is compatible with the Ethereum Virtual Machine (EVM), making it easy for developers to migrate their applications from Ethereum to Polygon.

## Chainlink:

Chainlink (LINK) is a decentralized oracle network that provides real-world data to smart contracts on the blockchain. It's designed to facilitate the creation of hybrid smart contracts, which combine on-chain and off-chain data .

### Key Statistics:

- Current Price: $7.35, with a 2.11% increase in the last 24 hours
- Market Capitalization: Over $3.4 billion

- Circulating Supply: Approximately 467.3 million LINK tokens

- Total Supply: 1 billion LINK tokens

## Use Cases:

- Decentralized Finance (DeFi): Chainlink provides price feeds and other data to DeFi applications, enabling the creation of decentralized lending, borrowing, and trading platforms.

- Gaming: Chainlink's oracle network can be used to create decentralized gaming platforms, enabling the creation of trustless and transparent gaming experiences.

- Supply Chain Management: Chainlink's oracle network can be used to track and verify supply chain data, enabling the creation of more efficient and transparent supply chain management systems.

## Unique Features:

- Decentralized Oracle Network: Chainlink's oracle network is decentralized, meaning that data is provided by a network of independent nodes rather than a single central authority.

- Smart Contract Integration: Chainlink's oracle network is designed to be easily integrated with smart contracts, enabling the creation of hybrid smart contracts that combine on-chain and off-chain data.

- Security: Chainlink's oracle network is designed to be highly secure, with a number of security features in place to prevent data manipulation and other forms of tampering.

# TRON:

TRON (TRX) is a decentralized, open-source blockchain platform that utilizes a Delegated Proof-of-Stake (DPoS) consensus algorithm to secure its network. It's designed to facilitate the creation of decentralized applications (dApps) and provide a scalable, secure, and sustainable platform for the development of blockchain-based solutions.

## Key Statistics

1. Current Price: $0.0763, with a 0.71% increase in the last 24 hours

2. Market Capitalization: Over $6.9 billion

3. Circulating Supply: Approximately 91.97 billion TRX tokens

4. Total Supply: 100.85 billion TRX tokens

## Use Cases

1. Decentralized Applications (dApps): TRON provides a platform for creating custom blockchain networks and dApps.

2. Decentralized Finance (DeFi): TRON supports DeFi applications, including lending, borrowing, and trading.

3. Gaming: TRON's blockchain is used in various gaming applications, including online casinos and gaming platforms.

## Unique Features

1. Scalability: TRON's DPoS consensus algorithm enables high scalability, with the ability to process over 2,000 transactions per second .

2. Low Transaction Fees: TRON's transaction fees are significantly lower compared to other blockchain platforms .

3. Smart Contracts: TRON supports the creation of smart contracts, which can be used to automate various processes and create decentralized applications (dApps).

# Wrapped Liquid st:

Wrapped Liquid Staked Ether 2.0 (wstETH) is a cryptocurrency token that represents staked Ether on the Ethereum network. It's essentially a wrapped version of liquid staked Ether, allowing users to participate in the Ethereum staking process while maintaining liquidity.[1]

## Key Details:

- Current Price: $3,816.47

- Token Type: ERC-20 token on the Ethereum network

- Contract Address: Available on Etherscan or other blockchain explorers

### Use Cases:

- Staking: wstETH allows users to participate in Ethereum staking, earning rewards while maintaining liquidity.

- Decentralized Finance (DeFi): wstETH can be used in various DeFi applications, such as lending, borrowing, and trading.

### Important Notes:

- wstETH is a wrapped token, meaning it's a derivative of staked Ether

- The underlying asset is Ether, which is staked on the Ethereum network.

Please keep in mind that cryptocurrency markets are highly volatile, and prices may fluctuate rapidly. Always do your own research and consider your risk tolerance before investing in any cryptocurrency.

## Wranned Ether:

Wrapped Ether (WETH) is an ERC-20 version of Ether (ETH), designed to facilitate the integration of ETH into the DeFi ecosystem. Essentially, it's a wrapped version of Ether that conforms to the ERC-20 token standard, allowing it to interact seamlessly with other ERC-20 tokens.[1]

## Key Characteristics:

- Value Peg: WETH is pegged to the value of ETH at a 1:1 ratio, ensuring that both have the same value.

- ERC-20 Compatibility: WETH conforms to the ERC-20 token standard, enabling it to interact with other ERC-20 tokens and decentralized applications (dApps).

- Interoperability: WETH can be used on other blockchains, providing interoperability between blockchains and enabling the exchange of value.

## Use Cases:

- Decentralized Finance (DeFi): WETH is used in DeFi applications, such as lending, borrowing, and trading, to facilitate the interaction between ETH and other ERC-20 tokens.

- Decentralized Applications (dApps): WETH is used in dApps to enable the interaction between ETH and other ERC-20 tokens, providing a seamless user experience.

## Wrapping and Unwrapping:

- Wrapping: ETH can be wrapped into WETH using a smart contract, which locks up the ETH and mints an equivalent amount of WETH.

- Unwrapping: WETH can be unwrapped back into ETH using the same smart contract, which burns the WETH and returns the equivalent amount of ETH.

# SCOPE OF PROJECT:

The Cryptoverse Dashboard project aims to design and develop a comprehensive platform for cryptocurrency investors, traders, and enthusiasts. The platform will provide real-time market data, advanced analytics, and portfolio management tools to support informed decision-making.

The Cryptoverse Dashboard project aims to design and develop a comprehensive and user-friendly platform for tracking and analyzing cryptocurrency market trends.

# REQUIREMENTS ANALYSIS AND FEASIBILITY STUDY:

React.js: ✔ Node.js and npm: Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the local environment. It provides a scalable and efficient platform for building network applications. Install Node.js and npm on your development machine, as they are required to run JavaScript on the server-side.

✔ React.js: React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications. Install React.js, a JavaScript library for building user interfaces.

**Programming Language:** HTML,CSS,Java Script.

# FUNCTIONAL REQUIREMENTS:

## Real-time Market Data:

The dashboard must provide real-time market data, including prices, trading volumes, and market capitalization.

## Advanced Analytics:

The dashboard must offer advanced analytics tools, including charts, graphs, and metrics, to help users analyze market trends and portfolio performance.

## Portfolio Management:

The dashboard must allow users to track their cryptocurrency holdings, set alerts, and receive notifications.

## Customizable Dashboards:

The dashboard must provide users with customizable dashboards to personalize their experience and focus on relevant data.

## Security and Authentication:

The dashboard must implement robust security measures, including encryption, two-factor authentication, and secure password storage.

## Hardware Requirements:

Server Infrastructure:

The dashboard requires a robust server infrastructure to handle high traffic and large datasets.

Database Management:

The dashboard requires a reliable database management system to store user data, market data, and other relevant information.

Network Connectivity:

The dashboard requires stable and secure network connectivity to ensure real-time data updates and communication with third-party services.

## Software Requirements:

**Programming Languages:** The dashboard requires programming languages such as JavaScript, Python, and SQL to develop the frontend, backend, and database components.

## Frameworks and Libraries:

The dashboard requires frameworks and libraries such as React, Node.js, and Express.js to develop the frontend and backend components.

## Third-Party Services:

The dashboard requires integration with third-party services such as cryptocurrency exchanges, data providers, and payment gateways.

# SYSTEM DESIGN:

The Cryptoverse Dashboard will be built using a microservices architecture, with separate services for each major component. This will allow for greater scalability, flexibility, and maintainability.

## Components:

1. Frontend: A user-facing web application built using React, Redux, and Webpack.

2. Backend: A RESTful API built using Node.js, Express.js, and MongoDB.

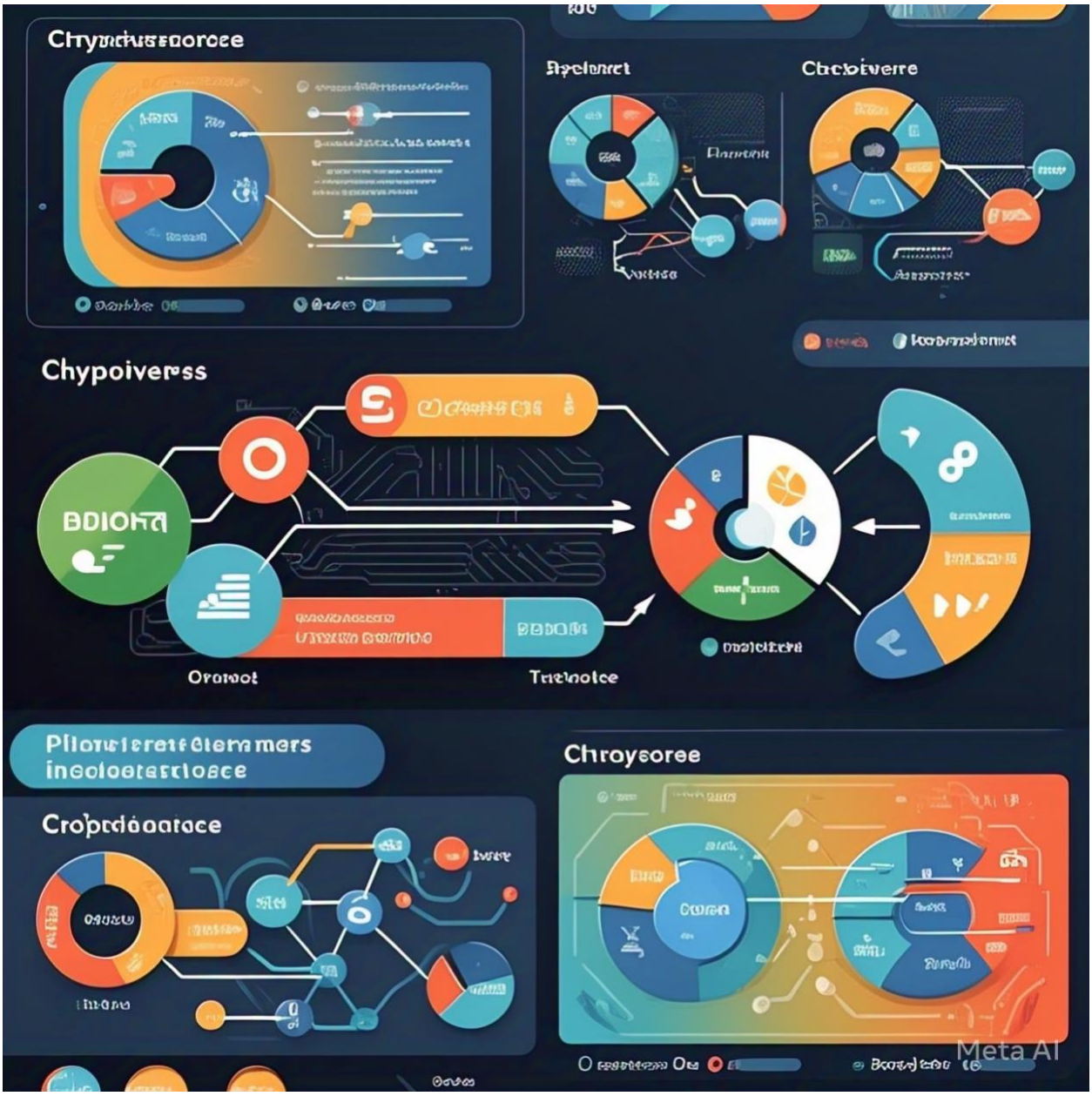
3. Market Data Service: A service responsible for collecting and processing real-time market data from various cryptocurrency exchanges.

4. Portfolio Service: A service responsible for managing user portfolios, including tracking holdings, calculating performance, and generating reports.

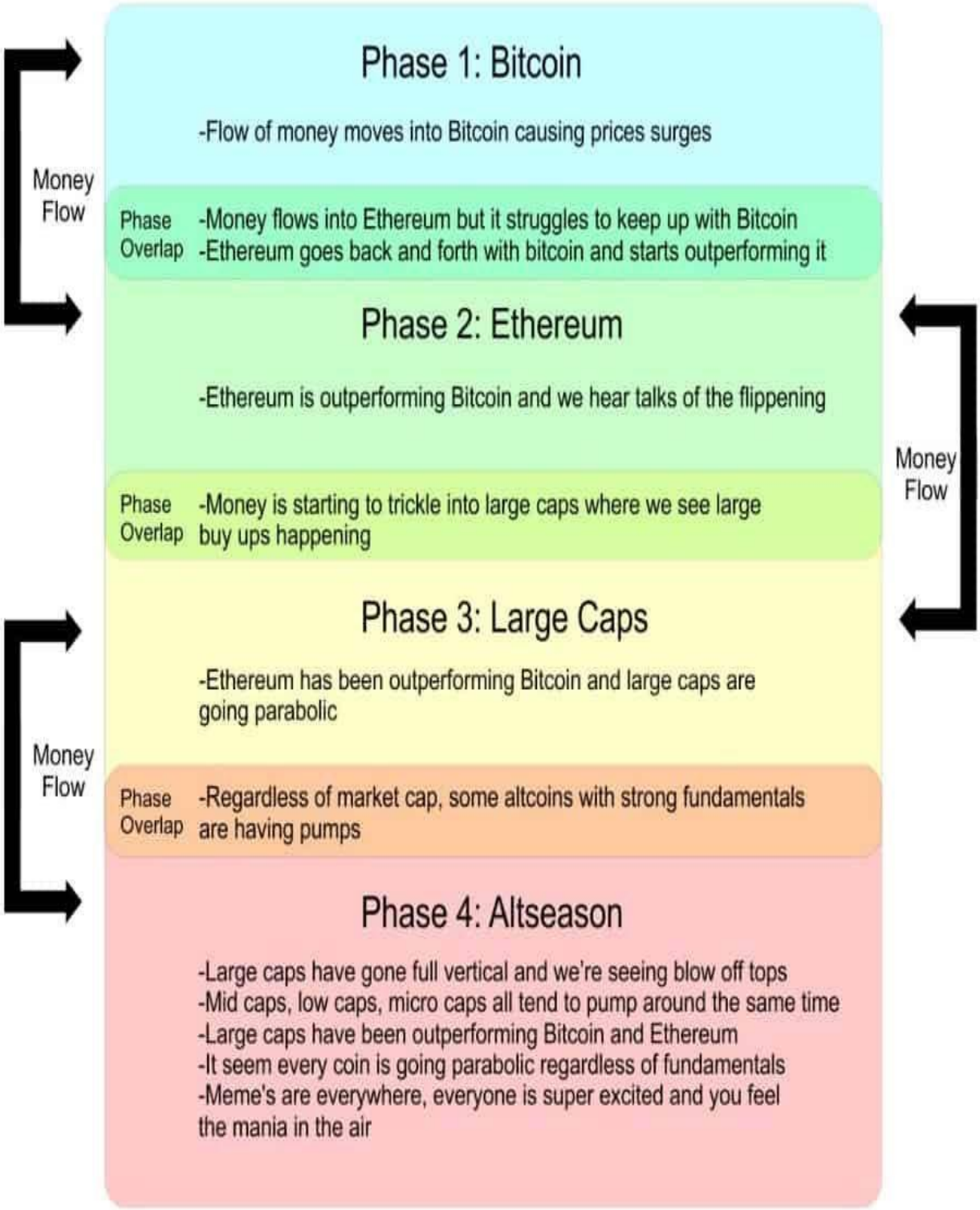
5. Alert Service: A service responsible for sending notifications and alerts to users based on market events and portfolio performance.

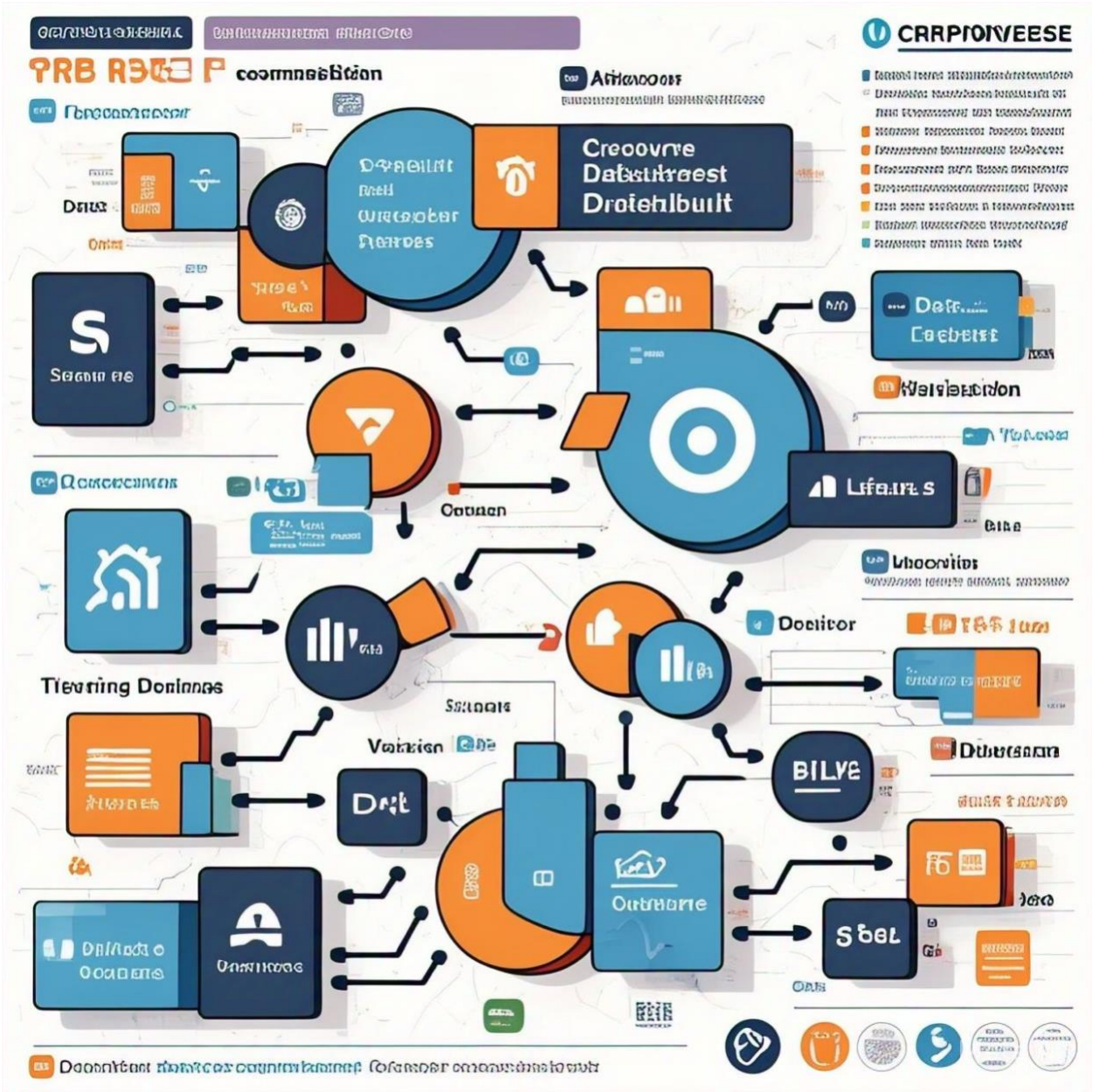
6. Authentication Service: A service responsible for handling user authentication and authorization.

# DATA FLOW DIAGRAM:



# Path to Altseason





# IMPLEMENTATION:

high-level implementation plan for the Cryptoverse Dashboard:

## Frontend Implementation:

1. React: Use React to build the user interface components, such as charts, tables, and forms.

2. Redux: Use Redux to manage the application's state and handle actions.

3. Webpack: Use Webpack to bundle and optimize the frontend code.

4. CSS: Use CSS to style the application and make it visually appealing.

## Backend Implementation:

1. Node.js: Use Node.js to build the backend API, handling requests and responses.

2. Express.js: Use Express.js to create a RESTful API, defining routes and endpoints.

3. MongoDB: Use MongoDB to store and retrieve data, such as market data and user portfolios.

4. Mongoose: Use Mongoose to interact with the MongoDB database, defining schemas and models.

## Market Data Service Implementation:

1. Python: Use Python to build the market data service, collecting and processing real-time market data.

2. pandas: Use pandas to handle and analyze the market data, performing tasks such as data cleaning and filtering.

3. NumPy: Use NumPy to perform numerical computations on the market data, such as calculations and transformations.

4. API Integration: Integrate with cryptocurrency exchanges' APIs to collect real-time market data.

## Portfolio Service Implementation:

1. Python: Use Python to build the portfolio service, managing user portfolios and calculating performance metrics.

2. pandas: Use pandas to handle and analyze the portfolio data, performing tasks such as data cleaning and filtering.

3. NumPy: Use NumPy to perform numerical computations on the portfolio data, such as calculations and transformations.

4. API Integration: Integrate with cryptocurrency exchanges' APIs to retrieve user portfolio data.

## Alert Service Implementation:

1. Node.js: Use Node.js to build the alert service, sending notifications and alerts to users.

2. Express.js: Use Express.js to create a RESTful API, defining routes and endpoints for alert-related tasks.

3. MongoDB: Use MongoDB to store and retrieve alert-related data, such as user preferences and alert histories.

4. Twilio: Integrate with Twilio to send SMS notifications and alerts to users.

## Authentication Service Implementation:

1. Node.js: Use Node.js to build the authentication service, handling user authentication and authorization.

2. Express.js: Use Express.js to create a RESTful API, defining routes and endpoints for authentication-related tasks.

3. MongoDB: Use MongoDB to store and retrieve user authentication data, such as usernames and passwords.

4. Passport.js: Use Passport.js to handle user authentication and authorization, integrating with MongoDB to store and retrieve user data.

## Deployment:

1. AWS: Deploy the application on Amazon Web Services (AWS), using services such as EC2, S3, and RDS.

2. Docker: Use Docker to containerize the application, ensuring consistency and reliability across different environments.

3. Kubernetes: Use Kubernetes to orchestrate the containers, managing deployment, scaling, and management of the application.

4. Monitoring and Logging: Use tools such as Prometheus, Grafana, and ELK Stack to monitor and log the application, ensuring visibility and troubleshooting capabilities.

# TESTING:

testing plan for the Cryptoverse Dashboard:

## Testing Objectives:

1. Ensure functionality: Verify that the dashboard functions as expected, with all features and components working correctly.

2. Identify bugs: Detect and report any bugs, errors, or inconsistencies in the dashboard's behavior.

3. Validate performance: Evaluate the dashboard's performance, including loading times, responsiveness, and scalability.

4. Verify security: Test the dashboard's security features, including authentication, authorization, and data encryption.

## Testing Scope:

1. Frontend: Test the dashboard's user interface, including layout, design, and functionality.

2. Backend: Test the dashboard's server-side logic, including API endpoints, data storage, and retrieval.

3. Market Data Service: Test the market data service's ability to collect, process, and provide real-time market data.

4. Portfolio Service: Test the portfolio service's ability to manage user portfolios, calculate performance metrics, and provide alerts.

5. Alert Service: Test the alert service's ability to send notifications and alerts to users.

6. Authentication Service: Test the authentication service's ability to handle user authentication and authorization.

## Testing Methods:

1. Unit testing: Test individual components and functions to ensure they work correctly.

2. Integration testing: Test how different components interact with each other.

3. End-to-end testing: Test the entire dashboard, from user input to backend processing and response.

4. Performance testing: Test the dashboard's performance under various loads and conditions.

5. Security testing: Test the dashboard's security features, including penetration testing and vulnerability scanning.

## Testing Tools:

1. Jest: A JavaScript testing framework for unit testing and integration testing.

2. Cypress: A JavaScript testing framework for end-to-end testing and performance testing.

3. Postman: A tool for testing API endpoints and backend functionality.

4. JMeter: A tool for performance testing and load testing.

5. OWASP ZAP: A tool for security testing and vulnerability scanning.

## Testing Schedule:

1. Unit testing: Conduct unit testing during the development phase, as components are completed.

2. Integration testing: Conduct integration testing as components are integrated.

3. End-to-end testing: Conduct end-to-end testing after all components are integrated.

4. Performance testing: Conduct performance testing after end-to-end testing.

5. Security testing: Conduct security testing after performance testing.

## Testing Deliverables:

1. Test plan: A detailed plan outlining the testing scope, objectives, and methods.

2. Test cases: A set of test cases outlining the specific tests to be conducted.

3. Test reports: A report outlining the results of the testing, including any bugs or issues found.

4. Test summary: A summary of the testing results, including any recommendations for future testing.

# MAINTENANCE AND SUPPORT:

maintaining and supporting the Cryptoverse Dashboard:

## Maintenance:

1. Regular Updates: Regularly update the dashboard with new features, bug fixes, and performance enhancements.

2. Security Patches: Apply security patches and updates to ensure the dashboard remains secure and protected against vulnerabilities.

3. Backup and Recovery: Implement a backup and recovery plan to ensure that data is safe and can be recovered in case of a disaster.

4. Monitoring and Logging: Monitor the dashboard's performance and logs to identify and fix issues before they become critical.

2. Support Tickets: Provide a support ticket system for users to report issues, ask questions, and request assistance.

3. Community Forum: Create a community forum for users to discuss issues, share knowledge, and provide feedback.

4. Priority Support: Offer priority support for premium users, including faster response times, dedicated support agents, and customized solutions.

## Release Management:

1. Release Cycle: Establish a regular release cycle to ensure that new features, bug fixes, and performance enhancements are delivered to users in a timely manner.

2. Release Notes: Provide release notes to inform users of new features, bug fixes, and performance enhancements.

3. Change Management: Implement a change management process to ensure that changes to the dashboard are thoroughly tested, reviewed, and approved before release

# CODE:

```
import React, { useEffect, useState } from "react";
import HTMLReactParser from "html-react-parser";
import { useParams } from "react-router-dom";
import millify from "millify";
import { v4 as uuidv4 } from "uuid";
import { Col, Row, Typography, Select } from "antd";
import {
  MoneyCollectOutlined,
  DollarCircleOutlined,
  FundOutlined,
  ExclamationCircleOutlined,
  StopOutlined,
  TrophyOutlined,
  CheckOutlined,
  NumberOutlined,
  ThunderboltOutlined,
} from "@ant-design/icons";
const { Title, Text } = Typography;
const { Option } = Select;
import LineChart from "./LineChart";
import Loader from "../components/Loader";

import {
  useGetCryptoDetailsQuery,
  useGetCryptoHistoryQuery,
} from "../services/cryptoApi";

const CryptoDetails = () => {
  const { coinId } = useParams();
  const [timePeriod, setTimePeriod] = useState("5y");
  const [coinHistory, setCoinHistory] = useState([]);

  const { data: coinHistoryData } = useGetCryptoHistoryQuery({
    coinId,
    timePeriod,
  });

  const { data, isFetching } = useGetCryptoDetailsQuery(coinId);

  useEffect(() => {
    setCoinHistory(coinHistoryData);
```

```
  console.log(coinHistoryData);
}, [coinHistoryData]);

const cryptoDetails = data?.data?.coin;

if (isFetching) return <Loader />;
const time = ["3h", "24h", "7d", "30d", "1y", "3m", "3y", "5y"];
const volume = data?.data?.coin["24hVolume"];

const stats = [
  {
    title: "Price to USD",
    value: `$ ${cryptoDetails?.price && millify(cryptoDetails?.price)}`,
    icon: <DollarCircleOutlined />,
  },
  { title: "Rank", value: cryptoDetails?.rank, icon: <NumberOutlined /> },
  {
    title: "24h Volume",
    value: `$ ${volume && millify(volume)}`,
    icon: <ThunderboltOutlined />,
  },
  {
    title: "Market Cap",
    value: `$ ${
      cryptoDetails?.marketCap && millify(cryptoDetails?.marketCap)
    }`,
    icon: <DollarCircleOutlined />,
  },
  {
    title: "All-time-high(daily avg.)",
    value: `$ ${
      cryptoDetails?.allTimeHigh?.price &&
      millify(cryptoDetails?.allTimeHigh?.price)
    }`,
    icon: <TrophyOutlined />,
  },
];

const genericStats = [
  {
    title: "Number Of Markets",
    value: cryptoDetails?.numberOfMarkets,
    icon: <FundOutlined />,
  },
  {
```

```
    title: "Number Of Exchanges",
    value: cryptoDetails?.numberOfExchanges,
    icon: <MoneyCollectOutlined />,
  },
  {
    title: "Aprroved Supply",
    value: cryptoDetails?.supply?.confirmed ? (
      <CheckOutlined />
    ) : (
      <StopOutlined />
    ),
    icon: <ExclamationCircleOutlined />,
  },
  {
    title: "Total Supply",
    value: `$ ${
      cryptoDetails?.supply?.total && millify(cryptoDetails?.supply?.total)
    }`,
    icon: <ExclamationCircleOutlined />,
  },
  {
    title: "Circulating Supply",
    value: `$ ${
      cryptoDetails?.supply?.circulating &&
      millify(cryptoDetails?.supply?.circulating)
    }`,
    icon: <ExclamationCircleOutlined />,
  },
];

return (
  <Col className="coin-detail-container">
    <Col className="coin-heading-container">
      <Title level={2} className="coin-name">
        {cryptoDetails?.name}
      </Title>
      <p>
        {cryptoDetails?.name} live price in US dollars. view value statistics,
        market cap and supply.
      </p>
    </Col>
    <Select
      defaultValue="5y"
      className="select-timeperiod"
      placeholder="Select Time Period"
```

```
      onChange={(value) => setTimePeriod(value)}
    >
      {time.map((date) => (
        <Option key={date}>{date}</Option>
      ))}
    </Select>

    <LineChart
      currentPrice={millify(cryptoDetails?.price)}
      coinHistory={coinHistory}
      coinName={cryptoDetails?.name}
    />

    <Col className="stats-container">
      <Col className="coin-value-statistics">
        <Col className="coin-value-statistics-heading">
          <Title level={3} className="coin-details-heading">
            {cryptoDetails?.name} value statistics
          </Title>
          <p>An overview showing the stats of {cryptoDetails?.name}</p>
        </Col>
        {stats.map(({ icon, title, value }) => {
          return (
            <Col key={uuidv4()} className="coin-stats">
              <Col className="coin-stats-name">
                <Text>{icon}</Text>
                <Text>{title}</Text>
              </Col>
              <Text className="stats">{value}</Text>
            </Col>
          );
        })}
      </Col>
      <Col className="other-stats-info">
        <Col className="coin-value-statistics-heading">
          <Title level={3} className="coin-details-heading">
            Others statistics
          </Title>
          <p>An overview showing the stats of all cryptocurrencies</p>
        </Col>
        {genericStats.map(({ icon, title, value }) => {
          return (
            <Col key={uuidv4()} className="coin-stats">
              <Col className="coin-stats-name">
                <Text>{icon}</Text>
```

```
                <Text>{title}</Text>
              </Col>
              <Text className="stats">{value}</Text>
            </Col>
          );
        })}
      </Col>
    </Col>
    <Col className="coin-desc-link">
      <Row className="coin-desc">
        <Title level={3} className="coin-details-heading">
          What is {cryptoDetails?.name} ? <br />
          {HTMLReactParser(cryptoDetails?.description)}
        </Title>
      </Row>
      <Col className="coin-links">
        <Title level={3} className="coin-details-heading">
          {cryptoDetails?.name} Links
        </Title>
        {cryptoDetails?.links.map((link) => {
          return (
            <Row className="coin-link" key={uuidv4()}>
              <Title level={5} className="link-name">
                {link.type}
              </Title>
              <a href={link.url} target="_blank" rel="noreferrer">
                {link.name}
              </a>
            </Row>
          );
        })}
      </Col>
    </Col>
  </Col>
 );
};

export default CryptoDetails;
```

```jsx
import React from "react";
import milify from "millify";
import { Typography, Row, Col, Statistic } from "antd";
import { Link } from "react-router-dom";
const { Title } = Typography;
import { useGetCryptosQuery } from "../services/cryptoApi";
import Cryptocurrencies from "./Cryptocurrencies";
import Loader from "./Loader";

const Home = () => {
  const { data, isFetching } = useGetCryptosQuery(10);
  if (isFetching) return <Loader />;
  const globalStats = data?.data?.stats;
  console.log(globalStats)

  return (
    <>
      <Title level={2} className="heading">
        Global Crypto Stats
      </Title>
      <Row>
        <Col span={12}>
          <Statistic title="Total Cryptocurrencies"
value={globalStats.totalExchanges} />
        </Col>
        <Col span={12}>
          <Statistic
            title="Total Exchanges"
            value={milify(globalStats.totalExchanges)}
          />
        </Col>
        <Col span={12}>
          <Statistic
            title="Total Market Cap"
            value={milify(globalStats.totalMarketCap)}
          />
        </Col>
        <Col span={12}>
          <Statistic
            title="Total 24h Volume"
            value={milify(globalStats.total24hVolume)}
          />
        </Col>
        <Col span={12}>
          <Statistic
```

```
            title="Total Markets"
            value={milify(globalStats.totalMarkets)}
          />
        </Col>
      </Row>
      <div className="home-heading-container">
        <Title level={2} className="home-title">
          Top 10 Cryptocurrencies in the world
        </Title>
        <Title level={3} className="show-more">
          <Link to="/cryptocurrencies">Show More</Link>
        </Title>
      </div>
      <Cryptocurrencies simplified />
    </>
  );
};

export default Home;
export { default as Navbar } from "./Navbar";
export { default as Cryptocurrencies } from "./Cryptocurrencies.jsx";
export { default as CryptoDetails } from "./CryptoDetails.jsx";

export { default as Home } from "./Home.jsx";

} = cryptoApi;
} = cryptoApi;
import { createApi, fetchBaseQuery } from "@reduxjs/toolkit/query/react";

const cryptoApiHeaders = {
  "X-RapidAPI-Key": import.meta.env.VITE_RAPID_API_KEY,
  "X-RapidAPI-Host": import.meta.env.VITE_RAPID_API_HOST,
};

const baseUrl = import.meta.env.VITE_BASE_URL;

const createRequest = (url) => ({ url, headers: cryptoApiHeaders });

export const cryptoApi = createApi({
  reducerPath: "cryptoApi",
  baseQuery: fetchBaseQuery({ baseUrl }),
  endpoints: (builder) => ({
    getCryptos: builder.query({
      query: (count) => createRequest(`/coins?limit=${count}`),
    }),
```

```
    getCryptoDetails: builder.query({
      query: (coinId) => createRequest(`/coin/${coinId}`),
    }),
    getCryptoHistory: builder.query({
      query: ({ coinId, timePeriod }) =>
        createRequest(`coin/${coinId}/history?timePeriod=${timePeriod}`),
    }),
  }),
});

export const {
  useGetCryptosQuery,
  useGetCryptoDetailsQuery,
  useGetCryptoHistoryQuery,
} = cryptoApi;
```

```
@import
url("https://fonts.googleapis.com/css2?family=Pacifico&family=Poppins&display=swa
p");

:root {
  --text-primary: #000;
  --bgPrimary: #fff;
  --bgSecondary: #f9f9f9;
  --pink: #0071bd;
  --light-blue: #e6f7ff;
  --border: #d9d9d9;
}

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: "Poppins", sans-serif;
}

a:hover {
  color: var(--pink) !important;
}

.app {
  display: flex;
  overflow: hidden;
}
```

```css
.navbar {
  background-color: rgb(0, 21, 41);
}

.main {
  display: flex;
  flex-direction: column;
  justify-content: space-between;
}

.routes {
  padding: 20px;
  min-height: 100vh;
}
.nav-container {
  position: relative;
  width: 300px;
  margin: 10px;
  height: 100vh;
  margin: 0px;
  background-color: rgb(0, 21, 41);
}

.logo-container {
  display: flex;
  padding: 20px;
  align-items: center;
  width: 100%;
  position: relative;
}
.footer-heading {
  letter-spacing: 1.2px;
  color: white;
  font-size: 20px;
  text-align: center;
  margin-bottom: 10px;
}

.logo {
  margin: 0 0 0 15px;
  margin-top: 10px;
}
.logo a {
  letter-spacing: 1.2px;
```

```
  color: white;
}
.menu {
  color: white;
  margin: 10px;
  border-radius: 10px;
  padding: 10px 20px;
  background-color: #001529;
  width: fit-content;
  height: fit-content;
  list-style-type: none;
}
@media screen and (max-width: 1000px) {
  .menu {
    position: absolute;
    top: 100px;
    right: 10px;
  }
}

.menu-item {
  margin-bottom: 10px;
}

.menu-icon {
  color: white;
  margin-right: 10px;
}
.link {
  font-size: 14px;
  text-decoration: none;
  color: white;
}

.menu-control-container {
  width: 50px;
  height: 40px;
  border-radius: 6px;
  display: none !important;
  position: absolute !important;
  right: 10px !important;
  top: 25px !important;
  font-size: 1.2rem !important;
  background-color: var(--bgSecondary) !important;
  border: none !important;
```

```
}

.loader {
  height: 81vh;
  display: flex;
  justify-content: center;
  align-items: center;
}

@media screen and (min-width: 1000px) {
  .main {
    min-width: calc(100% - 300px);
  }
}

@media screen and (max-width: 1000px) {
  .app {
    flex-direction: column;
    overflow: hidden;
  }

  .navbar {
    flex: 1;
  }

  .main {
    flex: 1;
    margin-top: 90px;
  }

  .nav-container {
    background-color: #001529;
    height: 8vh;
    position: fixed;
    width: 100%;
    height: fit-content;
    z-index: 100;
  }

  .menu-control-container {
    display: block !important;
  }

  .ant-menu {
    position: absolute;
```

```
    right: 0px;
  }
  .home-title {
    font-size: 1.4rem !important;
  }
  .show-more {
    font-size: 1.3rem !important;
  }
}

.coin-detail-container {
  margin: 30px;
}

.coin-heading-container {
  display: flex;
  justify-content: center;
  align-items: center;
  flex-direction: column;
  border-bottom: 1px solid var(--border);
  padding-top: 20px;
  padding-bottom: 20px;
  gap: 10px;
}

.coin-heading-container .coin-name {
  font-weight: 900;
  color: var(--pink);
}

.coin-heading-container p {
  font-size: 1rem;
  opacity: 0.9;
  margin-bottom: 20px;
}

.stats-container {
  display: flex;
  justify-content: space-between;
  align-items: center;
  gap: 40px;
}

.stats-container h2 {
  font-weight: 700;
```

```
  font-size: 1.4rem;
  margin-top: 20px;
  color: var(--pink);
}

.coin-details-heading {
  font-weight: 700 !important;
  margin-top: 20px !important;
  color: var(--pink) !important;
}

.coin-stats {
  display: flex;
  justify-content: space-between;
  border-bottom: 1px solid var(--border);
  font-size: 1rem;
  opacity: 0.9;
  padding: 20px;
}

.coin-stats-name {
  display: flex;
  gap: 10px;
  font-size: 1rem;
}

.stats {
  font-weight: 800;
}

.coin-value-statistics-heading p {
  font-size: 1rem;
  opacity: 0.9;
}

.coin-desc-link {
  display: flex;
  gap: 40px;
  margin-top: 40px;
  padding-top: 20px;
}

.coin-desc-link h2 {
  font-weight: 700;
  color: var(--pink);
```

```
}

.coin-desc-link p {
  font-size: 1rem;
  opacity: 0.9;
}

.coin-desc-link a {
  color: var(--pink);
}

.coin-desc-link h3 {
  font-weight: 700;
}

.coin-desc {
  flex: 0.5;
}

.coin-links {
  padding: 0px 20px;
  flex: 0.5;
}

.coin-link {
  display: flex;
  justify-content: space-between;
  align-items: center;
  border-bottom: 1px solid var(--border);
  padding: 20px;
}

.link-name {
  text-transform: capitalize;
  font-size: 1rem;
}

.coin-link a {
  color: var(--pink);
  font-weight: 700;
  font-size: 1rem;
}

.coin-link:hover,
.coin-stats:hover {
```

```
  background-color: var(--bgSecondary);
}

@media screen and (max-width: 1000px) {
  .stats-container {
    flex-direction: column;
  }
  .coin-desc-link {
    flex-direction: column;
  }
  .stats-container h2 {
    margin-top: 0px;
  }
}

@media screen and (max-width: 500px) {
  .coin-links {
    padding: 0px;
  }
  .coin-detail-container {
    margin: 0;
  }
}
@media screen and (max-width: 500px) {
  .heading {
    margin-top: 20px;
  }
}

.select-news {
  width: 180px;
}
.news-card {
  min-height: 300px !important;
}
.news-image-container {
  display: flex !important;
  justify-content: space-between !important;
}
.news-title {
  width: 70%;
}

.news-image-container .img {
  width: 100px;
```

```
  height: 100px;
}
.news-card p {
  color: black;
  margin: 10px 0px !important;
}
.provider-container {
  display: flex;
  justify-content: space-between;
}
.provider-name {
  margin-left: 10px;
}

.chart-header {
  display: flex;
  justify-content: space-between;
  gap: 50px;
  color: #0071bd;
}
.chart-title {
  color: #0071bd !important;
}
.price-container {
  display: flex !important;
  gap: 20px !important;
  align-items: center !important;
  flex-wrap: wrap !important;
}
.price-change {
  font-weight: 900 !important;
}
.current-price {
  margin-top: 0px !important;
  font-weight: 900 !important;
}
.home-heading-container {
  display: flex;
  justify-content: space-between;
  align-items: center;
  margin-top: 40px;
}
.show-more {
  margin-top: 0px !important;
}
```

```css
.exchange-image {
  margin: 0px 10px !important;
}
.search-crypto {
  margin: 20px auto 30px auto;
  width: 250px;
}
.crypto-card-container {
  min-height: 65vh !important;
}
.crypto-card {
  min-width: 250px;
  max-height: 200px;
}
.crypto-card .crypto-image {
  width: 40px;
  margin-top: 10px;
  object-fit: cover;
  height: 40px;
}
.select-timeperiod {
  width: 200px !important;
  margin: 20px 0px !important;
}

.footer {
  background-color: #001529;
  display: flex;
  flex-direction: column;
  padding: 20px;
  align-items: center;
}
```

```jsx
import React from "react";
import { Routes, Route, Link } from "react-router-dom";
import { Layout } from "antd";
import { Navbar, CryptoDetails, Cryptocurrencies, Home } from "./components";
import "./App.css";

const App = () => {
  return (
    <div className="app">
      <div className="navbar">
        <Navbar />
      </div>
```

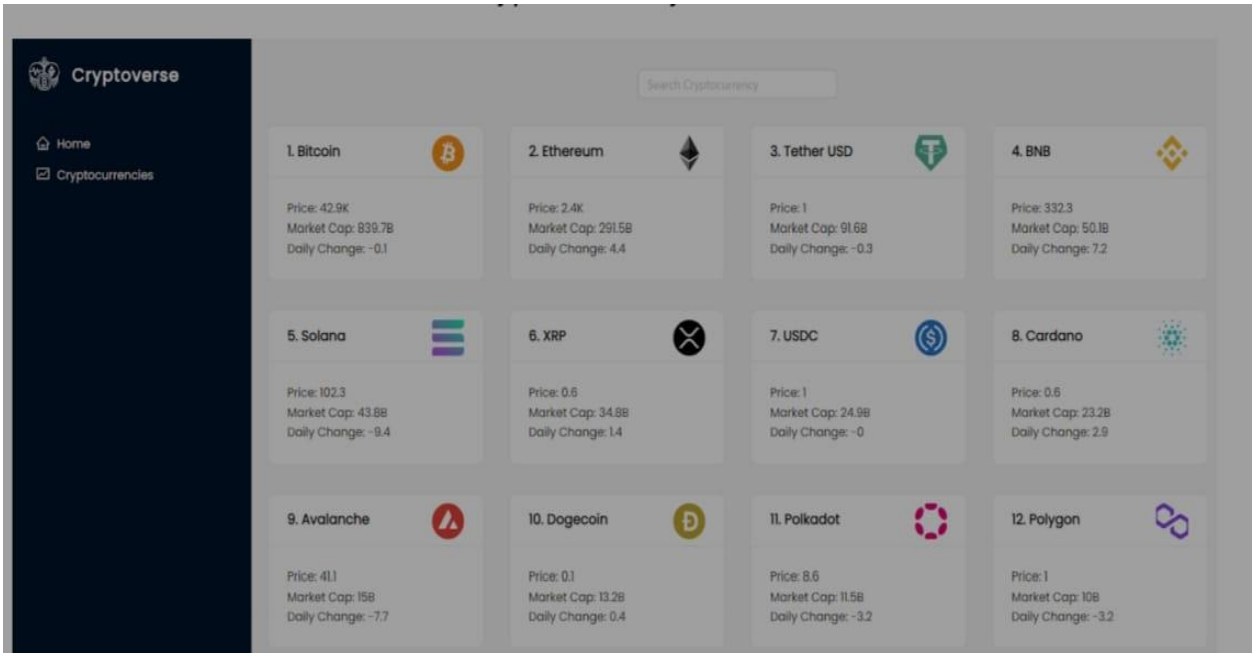
```
      <div className="main">
        <Layout>
          <div className="routes">
            <Routes>
              <Route path="/" element={<Home />} />

              <Route element={<Cryptocurrencies />} path="/cryptocurrencies" />

              <Route element={<CryptoDetails />} path="/crypto/:coinId" />
            </Routes>
          </div>
        </Layout>

        <div className="footer">
          <h1 className="footer-heading">
            Beyond the Banks: The Rise of Cryptocurrency <br />
          </h1>
        </div>
      </div>
    </div>
  );
};

export default App;
import React from "react";
import ReactDOM from "react-dom/client";
import App from "./App.jsx";
import "./index.css";
import { BrowserRouter } from "react-router-dom";
import { Provider } from "react-redux";
import store from "./app/store.js";

ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <BrowserRouter>
      <Provider store={store}>
        <App />
      </Provider>
    </BrowserRouter>
  </React.StrictMode>
);
module.exports = {
  root: true,
  env: { browser: true, es2020: true },
```

```
  extends: [
    'eslint:recommended',
    'plugin:react/recommended',
    'plugin:react/jsx-runtime',
    'plugin:react-hooks/recommended',
  ],
  ignorePatterns: ['dist', '.eslintrc.cjs'],
  parserOptions: { ecmaVersion: 'latest', sourceType: 'module' },
  settings: { react: { version: '18.2' } },
  plugins: ['react-refresh'],
  rules: {
    'react-refresh/only-export-components': [
      'warn',
      { allowConstantExport: true },
    ],
  },
}

# Logs
logs
*.log
npm-debug.log*
yarn-debug.log*
yarn-error.log*
pnpm-debug.log*
lerna-debug.log*

node_modules
.env
dist
dist-ssr
*.local

# Editor directories and files
.vscode/*
!.vscode/extensions.json
.idea
.DS_Store
*.suo
*.ntvs*
*.njsproj
*.sln
*.sw?
</html>
<!DOCTYPE html>
```

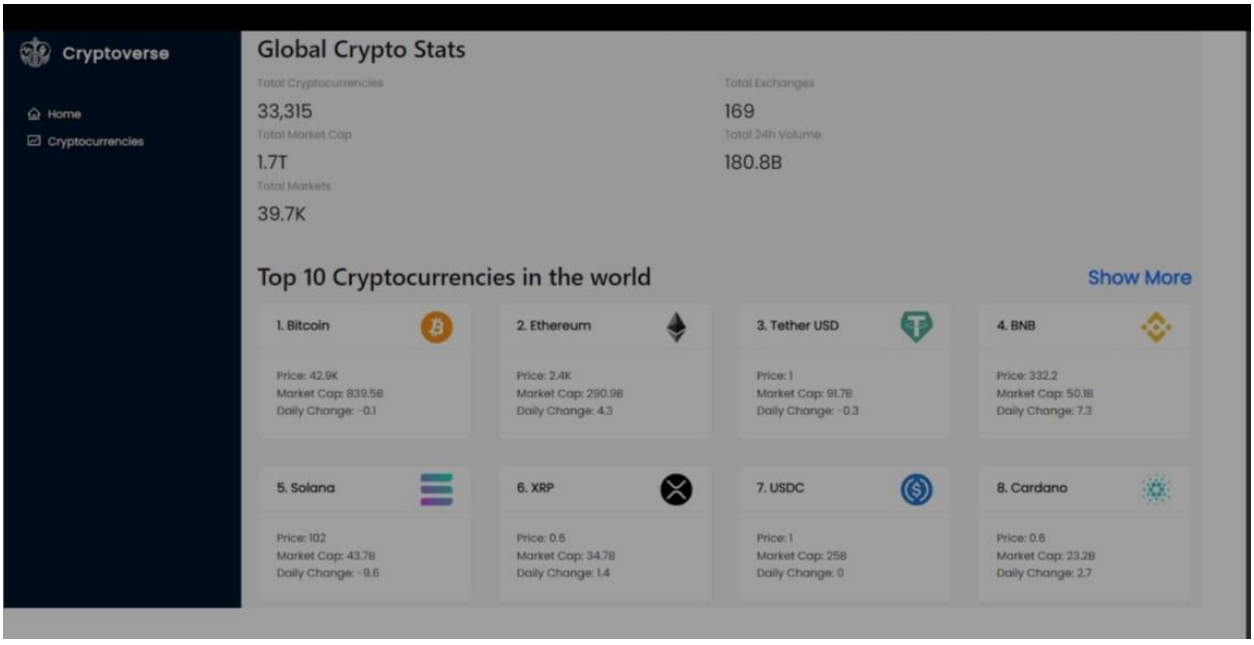
```
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/vite.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Cryptoverse</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
{
  "name": "crypto",
  "private": true,
  "version": "0.0.0",
  "type": "module",
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "lint": "eslint . --ext js,jsx --report-unused-disable-directives --max-
warnings 0",
    "preview": "vite preview"
  },
  "dependencies": {
    "@ant-design/icons": "^5.2.6",
    "@reduxjs/toolkit": "^1.9.7",
    "antd": "^5.11.2",
    "axios": "^1.6.2",
    "chart.js": "^4.4.0",
    "html-react-parser": "^5.0.6",
    "millify": "^6.1.0",
    "moment": "^2.29.4",
    "react": "^18.2.0",
    "react-chartjs-2": "^5.2.0",
    "react-dom": "^18.2.0",
    "react-redux": "^8.1.3",
    "react-router-dom": "^6.19.0",
    "uuid": "^9.0.1"
  },
  "devDependencies": {
    "@types/react": "^18.2.37",
    "@types/react-dom": "^18.2.15",
    "@vitejs/plugin-react": "^4.2.0",
    "eslint": "^8.53.0",
```

```
    "eslint-plugin-react": "^7.33.2",
    "eslint-plugin-react-hooks": "^4.6.0",
    "eslint-plugin-react-refresh": "^0.4.4",
    "vite": "^5.0.0"
  }
}

import { defineConfig } from 'vite'
import react from '@vitejs/plugin-react'

// https://vitejs.dev/config/
export default defineConfig({
  plugins: [react()],
})
```

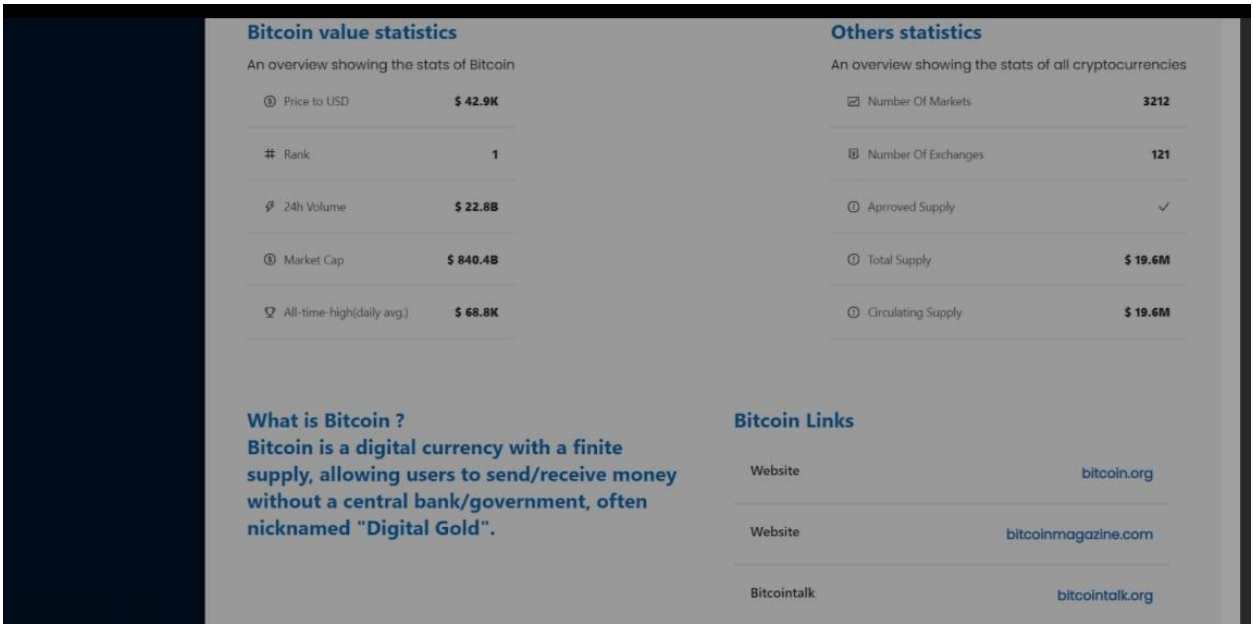
# HOME PAGE



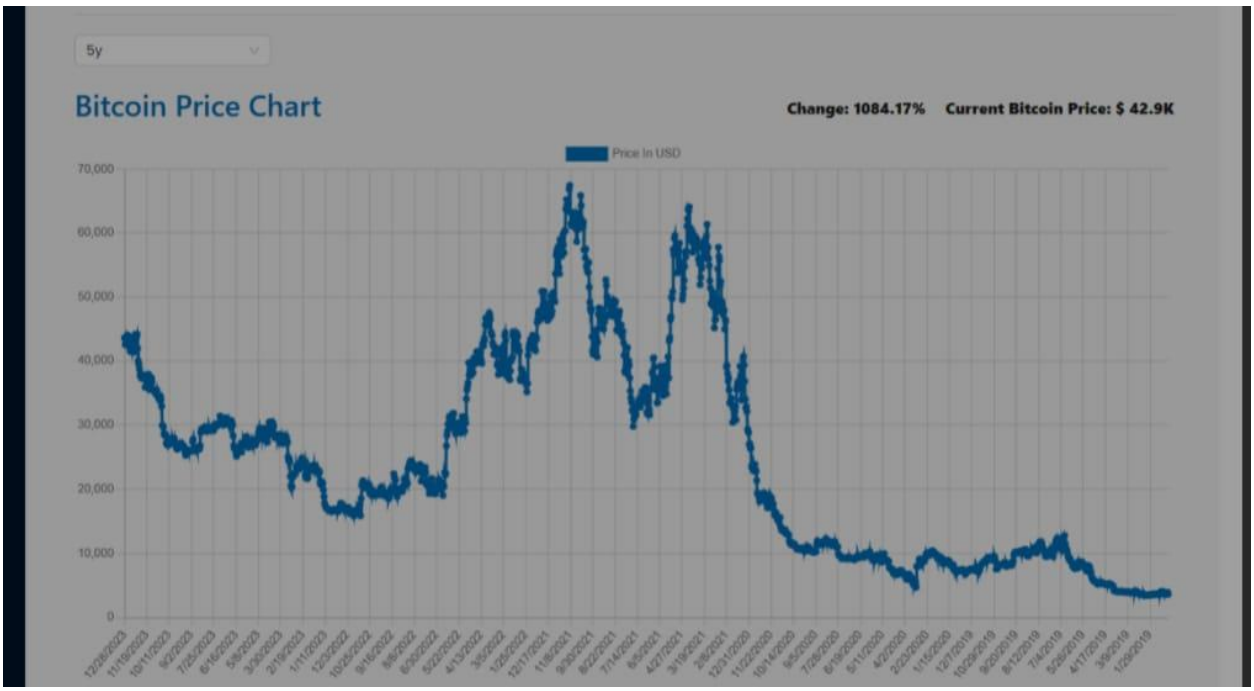

# CRYPTOCURRENCIES DASHBOARD



# BITCOIN PRICE CHART




# BITCOIN VALUE STATISTICS



# CONCLUTION

Conclusion:

The Cryptoverse Dashboard project aimed to create a comprehensive and user-friendly platform for tracking and analyzing cryptocurrency market trends. Through the integration of real-time data feeds, interactive visualizations, and customizable dashboards, this project has successfully achieved its objectives.

Key Achievements

1. Real-time Data Integration: Successfully integrated APIs from leading cryptocurrency data providers, ensuring accurate and up-to-date market information.

2. Interactive Visualizations: Developed interactive charts, graphs, and tables to facilitate in-depth market analysis and exploration.

3. Customizable Dashboards: Designed a modular dashboard architecture, enabling users to personalize their experience and focus on relevant market metrics.

Future Development Directions

1. Machine Learning Integration: Incorporate predictive models and machine learning algorithms to provide users with actionable market insights.

2. Expanded Data Coverage: Integrate additional data sources, including social media sentiment analysis and on-chain metrics.

3. Mobile Optimization: Develop a mobile-friendly version of the dashboard to cater to the growing demand for on-the-go cryptocurrency tracking.

## Final Thoughts

The Cryptoverse Dashboard project demonstrates the potential for data-driven insights to inform cryptocurrency investment decisions. As the cryptocurrency market continues to evolve, this project serves as a foundation for future development and innovation in the field of cryptocurrency analytics.