Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 863-871

CryptoNova: A Real-Time Cryptocurrency Analytics Dashboard

[1]Kapil Vishwakarma, [2]Pawan Kumar
[1]Student, [2]Assistant Professor
[1,2]Amity University Chhattisgarh
[1]Kapilvishwakarma755@gmail.com, [2]pkumar@rpr.amity.edu

Abstract

The rapidly evolving cryptocurrency market presents significant challenges due to its inherent high volatility and the growing demand for real-time analytical tools. Despite widespread interest from investors, researchers, and financial technologists, accessible and interpretable platforms that consolidate live market data with meaningful visual analytics remain scarce. To address this gap, we developed CryptoNova — an interactive cryptocurrency dashboard designed to simplify complex market data into a user-friendly and responsive interface. Built using React, TypeScript, and Vite, and integrated with the CoinGecko API, the system provides real-time insights including live prices, market capitalization metrics, percentage changes, and trend indicators. The primary objective of this project is to establish a scalable, resilient, and responsive platform for cryptocurrency tracking and analysis. The system incorporates fallback mechanisms to maintain functionality during API disruptions, ensuring consistent data availability under network-constrained conditions. Development began with a Streamlit-based prototype, which was subsequently replaced by a modern React-TypeScript architecture for improved performance, modularity, and user experience. CryptoNova includes key features such as a market overview dashboard, interactive historical price charts, top movers analysis with momentum tracking, and intelligent search functionality with real-time filtering. Performance evaluation shows efficient real-time data updates with an average rendering time of approximately 3.5 seconds across a range of devices and network conditions. This project contributes to the intersection of data science, financial technology, and human-computer interaction by combining real-time data collection, dynamic visualization, and analytical insights into a unified platform. It further demonstrates a scalable and adaptable architectural approach for building data-driven applications in dynamic and volatile environments.

Keywords: Cryptocurrency, Real-time Dashboard, React, TypeScript, CoinGecko API, Data Visualization, Market Analytics, Financial Technology, Vite, Streamlit

1. Introduction

Over the past decade, the cryptocurrency ecosystem has grown from a niche technological experiment into a global financial phenomenon encompassing thousands of digital assets with a combined market capitalization exceeding several trillion dollars. Unlike traditional equity markets governed by regulated exchanges and fixed trading hours, cryptocurrency markets operate continuously, 24 hours a day, seven days a week, across decentralized and geographically

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 863-871

distributed platforms [1]. This perpetual activity generates enormous streams of data: price ticks, volume records, order book updates, sentiment indices, and cross-asset correlations. For individual investors, analysts, and researchers, making sense of this data in real time is not just a matter of convenience, it can be the difference between informed decisions and costly errors. Yet, most existing platforms either cater to experienced traders through complex interfaces, or offer simplified views that sacrifice analytical depth [8]. This paper introduces CryptoNova, a web-based cryptocurrency analytics dashboard that bridges the gap between data richness and interpretive clarity. The platform is designed with three guiding principles: accessibility (information should be understandable to both novice and experienced users), resilience (the system must function reliably even during API disruptions), and performance (real-time updates should be delivered with minimal latency and rendering overhead) [2] [8]. The remainder of this paper is organized as follows. Section 2 reviews related work in financial dashboards and cryptocurrency analytics. Section 3 describes the system architecture and technology stack. Section 4 outlines the development methodology and iterative design process. Section 5 details the key features of CryptoNova. Section 6 presents performance evaluation results. Section 7 discusses findings and limitations. Section 8 concludes and identifies directions for future work.

2. Related Work

2.1 Cryptocurrency Market Analysis Platforms

A number of commercial platforms have established themselves as leaders in cryptocurrency data aggregation. CoinMarketCap, launched in 2013, was one of the earliest aggregators to offer standardized ranking of cryptocurrencies by market capitalization. CoinGecko, introduced in 2014, expanded this model by incorporating additional metrics such as developer activity, community engagement scores, and liquidity indicators. Both platforms offer public APIs that have become foundational infrastructure for third-party cryptocurrency applications [7]. Academic research in this space has explored cryptocurrency price prediction using machine learning models, sentiment analysis of social media feeds, and portfolio optimization under extreme volatility conditions. Notably, studies have leveraged LSTM (Long Short-Term Memory) networks, transformer-based architectures, and ensemble methods to forecast short-term price movements, with mixed results attributable to the inherently chaotic nature of crypto markets. [12] [13]

2.2 Financial Data Visualization

Effective visualization of financial data has been a research focus in both human-computer interaction and data science communities. Interactive candlestick charts, heat maps for sector performance, and sparklines for trend indication have all been studied in the context of decision support systems. Libraries such as D3.js and Chart.js have democratized sophisticated financial

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 863-871

charting within web browsers, enabling developers to embed rich interactivity without proprietary software dependencies. [3] [14]

Earlier dashboard implementations frequently relied on server-side rendering and periodic page refreshes, which introduced latency and degraded the user experience in fast-moving markets. The shift toward Single Page Application (SPA) architectures, particularly React-based systems, has enabled dashboards to update individual data components in real time without full page reloads, significantly improving responsiveness and usability. [9]

2.3 Rapid Prototyping with Streamlit

Streamlit emerged as a widely adopted tool in the data science community for building interactive data applications rapidly using Python. Its declarative syntax and automatic re-execution model allow analysts to go from raw data to visual interface within hours. However, Streamlit applications are fundamentally server-rendered and stateful in ways that limit their responsiveness for real-time streaming use cases and restrict frontend customization. CryptoNova's development trajectory, starting from Streamlit and transitioning to React, reflects a common pattern in the maturation of data applications from exploratory prototypes to production-ready systems. [2] [15]

3. System Architecture

3.1 Technology Stack

CryptoNova is implemented as a client-side Single Page Application built on the following technology stack:

Component	Technology / Library
Frontend Framework	React 18 with TypeScript
Build Tool	Vite 5
Charting Library	Recharts
Data Source	CoinGecko Public API v3
Styling	Tailwind CSS
State Management	React Hooks (useState, useEffect)
HTTP Client	Fetch API (native browser)

Table 1: Technology stack of CryptoNova platform. [1][9]

3.2 Data Flow Architecture

At the heart of CryptoNova's architecture is a unidirectional data flow model. When the application loads, it initiates asynchronous HTTP requests to the CoinGecko API endpoints to

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 863-871

retrieve current market data. This data is parsed, normalized, and stored in React component state. Periodic polling intervals, configurable by the developer, trigger re-fetches to ensure data freshness. All API interactions are encapsulated within custom React hooks, promoting separation of concerns and reusability[8] [9]. React's virtual DOM ensures that only the components whose underlying data has changed are re-rendered upon each update cycle. This selective update mechanism dramatically reduces unnecessary DOM manipulation, yielding smoother user experiences, especially on lower-powered mobile devices [9].

3.3 Fallback and Resilience Mechanisms

A distinctive architectural concern addressed in CryptoNova is resilience under API disruption. The CoinGecko free-tier API enforces rate limits, and transient network failures are common in browser-based deployments. To address this, the system implements the following fallback strategies:

- Cached Data Display: When an API call fails, the system serves the most recently fetched valid dataset rather than displaying an error state.[10]
- Graceful Error Boundaries: React error boundaries capture rendering exceptions at the component level, preventing application-wide crashes.[9]
- Retry Logic with Exponential Backoff: Failed API calls are automatically retried with increasing delay intervals to reduce load on the API server.[10]
- User-Facing Status Indicators: Visual indicators inform users when data may be stale or when a connection issue has been detected.[8]

4. Development Methodology

4.1 Phase 1: Streamlit Prototype

The project began with a rapid prototyping phase using Streamlit and Python. This approach enabled quick validation of core user experience concepts: what data fields are most useful, which chart types communicate price trends most intuitively, and how search and filtering should behave. The Streamlit prototype was deployed locally and demonstrated to a small group of potential users for feedback collection [2] [15]. While the prototype was effective as a proof of concept, its limitations became apparent quickly. Streamlit's re-execution model meant that user interactions triggered full Python script re-runs, producing noticeable delays. Customizing the visual appearance beyond Streamlit's built-in widgets required complex workarounds. These constraints motivated the transition to a dedicated frontend framework [2].

4.2 Phase 2: React-TypeScript Migration

Following the prototype phase, the development shifted to a React-TypeScript architecture scaffolded with Vite. TypeScript was chosen for its static type system, which catches category-

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 863-871

level errors at compile time rather than runtime, significantly improving code reliability in a codebase with complex nested data structures from API responses [9] [16]. Vite was selected as the build tool due to its use of native ES modules during development, resulting in near-instantaneous hot module replacement (HMR). For a project that required frequent iterative adjustments to visual components and data transformations, this speed advantage was practically significant [9].

4.3 Iterative Design and User-Centered Development

The dashboard layout and feature set were refined through multiple iterations informed by informal user feedback sessions. Early feedback revealed that users wanted quicker orientation to top-performing and worst-performing assets, which led to the development of the Top Movers module. Feedback also indicated a preference for percentage change indicators displayed with directional color coding (green for positive, red for negative) over raw numerical deltas alone [5] [17]. Accessibility considerations were incorporated progressively: sufficient color contrast ratios, keyboard-navigable components, and screen reader-compatible ARIA attributes were added as the interface stabilized. While a comprehensive accessibility audit was beyond the scope of this project, these foundational measures improve inclusivity for a broader user base [17].

5. Key Features of CryptoNova

5.1 Market Overview Dashboard

The market overview screen serves as the landing page of CryptoNova and provides a snapshot of the broader cryptocurrency market. It displays a ranked list of cryptocurrencies showing each asset's name, ticker symbol, current price in USD, 24-hour percentage change, 7-day percentage change, and market capitalization. Data is fetched from the CoinGecko /coins/markets endpoint, which supports pagination, currency conversion, and field selection parameters [7].

Each row in the market table is rendered as a React component that updates reactively when new data is received. Sparkline mini-charts, compact 7-day price trend lines, are optionally embedded within the table rows to provide immediate visual pattern recognition without requiring navigation to a dedicated chart page [3] [14].

5.2 Interactive Historical Price Charts

Clicking on any cryptocurrency from the market overview navigates to a detailed view featuring interactive historical price charts. Charts are rendered using the Recharts library, which provides a React-native charting solution built on D3.js primitives. Users can toggle between multiple time intervals, 24 hours, 7 days, 30 days, and 1 year, with each selection triggering a new API call to the CoinGecko market chart endpoint. [3] [14]

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 863-871

Recharts' responsive container component ensures that charts adapt fluidly to different screen sizes, from large desktop monitors to smaller smartphone displays. Tooltip overlays display precise price and timestamp information on hover or touch, enhancing data readability and user engagement. [9]

5.3 Top Movers Analysis

The Top Movers module identifies the cryptocurrency assets with the highest positive and negative 24-hour percentage changes within the tracked universe. This feature is particularly valuable for momentum traders and researchers interested in identifying short-term sentiment signals. The module ranks assets by absolute percentage change and displays the top five gainers and top five losers in a card-based layout with directional color coding.

5.4 Intelligent Search and Filtering

CryptoNova includes a search bar that filters the asset list in real time as users type, matching against both the full asset name and ticker symbol. The filter is implemented entirely client-side on the cached dataset, ensuring zero API calls and instantaneous response regardless of network conditions. Additionally, users can sort the asset list by any column (price, market cap, 24-hour change) in ascending or descending order. [8] [10]

6. Performance Evaluation

6.1 Metrics and Testing Methodology

CryptoNova's performance was evaluated across three device categories — desktop (Intel Core i7, 16 GB RAM), mid-range laptop (Intel Core i5, 8 GB RAM), and mobile (Qualcomm Snapdragon 720G, 6 GB RAM) — using Google Chrome with the DevTools Performance panel. Three primary metrics were recorded: Time to First Contentful Paint (FCP), Time to Interactive (TTI), and API-to-render latency (the duration from API call initiation to completed chart rendering). [10]

Device	FCP (s)	TTI (s)	API-to-Render (s)
Desktop	0.8	1.4	2.9
Laptop	1.1	2.0	3.4
Mobile	1.6	3.1	4.2
Average	1.2	2.2	3.5

Table 2: Performance metrics across device categories. FCP = First Contentful Paint, TTI = Time to Interactive. [10]

6.2 Analysis and Discussion of Results

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 863-871

The results demonstrate that CryptoNova delivers acceptable performance across all tested device categories. Desktop and laptop devices achieve FCP values well within the 1.8-second threshold recommended by Google Lighthouse for a "Good" rating. Mobile performance, while slightly elevated due to reduced CPU clock speeds and potential network throttling in test conditions, remains usable with an API-to-render time of 4.2 seconds. The overall average of 3.5 seconds for API-to-render is consistent with the performance envelope reported by comparable browser-based financial dashboards in the literature. [10] [18]

The primary latency bottleneck was found to be the CoinGecko API response time, which averaged 1.8 seconds for the market data endpoint under free-tier rate limits. React rendering overhead, by contrast, accounted for only 0.3–0.5 seconds across devices, reflecting the efficiency of the virtual DOM diffing mechanism. Bundle size optimization through Vite's tree-shaking and code-splitting features kept the initial JavaScript payload below 180 KB (gzipped), contributing to the relatively fast FCP measurements. [9] [16]

7. Discussion

7.1 Contributions to the Field

CryptoNova makes several contributions at the intersection of data science, software engineering, and financial technology. First, it provides a publicly accessible, open-architecture platform that researchers and educators can use or extend for cryptocurrency market analysis without incurring commercial licensing costs. Second, the project's documented migration from Streamlit to React offers a practical case study for data science practitioners navigating the common decision between rapid Python-based prototyping and production-grade JavaScript frontends. [2] [4]

Third, the fallback and resilience mechanisms developed for CryptoNova address a practical concern that is frequently underemphasized in academic dashboard implementations: what happens when the data source becomes unavailable? The strategies implemented here, cached data display, retry logic, and user-facing status indicators, are transferable to any application that depends on third-party APIs in production environments. [10] [8]

7.2 Limitations

The current implementation carries several limitations that should be acknowledged. The reliance on the CoinGecko free-tier API introduces rate limiting constraints that restrict how frequently data can be refreshed and how many assets can be tracked simultaneously. In high-frequency trading contexts or research requiring tick-level data, this would be insufficient. [7]

Furthermore, CryptoNova does not currently incorporate predictive analytics or machine learning-based price forecasting. The dashboard is a descriptive tool — it shows what has happened and what is happening now, but offers no prescriptive guidance on likely future price movements.

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 863-871

Integrating even simple statistical forecasting (e.g., exponential moving averages, Bollinger Bands) would meaningfully enhance the analytical value of the platform. [12] [13]

Finally, the current architecture processes all data in the browser, which is appropriate for the scale of data served by the CoinGecko API but would not be suitable for more computationally intensive analyses. Future iterations involving on-chain data, cross-exchange arbitrage detection, or portfolio optimization would require a backend processing layer. [11]

8. Conclusion and Future Work

This paper presented CryptoNova, a real-time cryptocurrency analytics dashboard built with React, TypeScript, Vite, and the CoinGecko API. The system successfully achieves its design objectives: it delivers live market data in a visually interpretable interface, functions resiliently under API disruption, and performs efficiently across device categories with an average rendering time of 3.5 seconds. The development trajectory, from Streamlit prototype to React production application, demonstrates a practical and well-documented pathway for evolving data science applications from exploratory tools to deployable products. [1] [2] [9]

Looking ahead, several directions for future work have been identified. Integrating a WebSocket-based data stream would eliminate the latency of periodic polling and enable true real-time tick-by-tick price display. Adding a Python-powered backend service for statistical analysis and forecasting would bridge the gap between the current descriptive dashboard and a prescriptive analytical platform. Support for user-defined portfolio tracking, with cost-basis calculation and return attribution, would greatly expand the practical utility of CryptoNova for active investors and researchers alike. [12] [11]

Ultimately, CryptoNova demonstrates that with thoughtful architectural choices and a user-centered design philosophy, it is possible to build accessible, resilient, and performant data-driven applications in one of the most volatile and data-intensive domains in modern finance. The platform and its underlying design patterns are intended to serve as a foundation that the research community and developer community alike can build upon. [4] [17]

References

[1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008.
[2] Streamlit Inc., “Streamlit: The fastest way to build data apps,” 2023. [3] M. Bostock, V. Ogievetsky, and J. Heer, “D³: Data-Driven Documents,” IEEE Trans. Vis. Comput. Graph., vol. 17, no. 12, pp. 2301–2309, 2011.
[4] S. Meiklejohn et al., “A Fistful of Bitcoins: Characterizing Payments Among Men with No Names,” in Proc. IMC ’13, 2013.

ISSN: 2584-1491 | www.iircj.org
Volume-4 | Issue-5 | May-2026 | Page 863-871

[5] B. Lim and S. Zohren, “Time-series Forecasting with Deep Learning: A Survey,” Phil. Trans. R. Soc. A, 2021.
[6] N. Gandal and H. Halaburda, “Can We Predict the Winner in a Market with Network Effects?” Games, vol. 7, no. 3, 2016.
[7] CoinGecko, “CoinGecko API Documentation v3,” 2024.
[8] J. Nielsen, Usability Engineering. Morgan Kaufmann, 1994.
[9] Facebook Inc., “React – A JavaScript Library for Building User Interfaces,” 2023.
[10] Google, “Lighthouse Performance Scoring,” 2024.
[11] V. Buterin, “A Next-Generation Smart Contract and Decentralized Application Platform,” 2014.
[12] T. Fischer and C. Krauss, “Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions,” Eur. J. Oper. Res., vol. 270, no. 2, 2018.
[13] Q. Wen et al., “Transformers in Time Series: A Survey,” in Proc. IJCAI, 2023.
[14] Recharts Team, “Recharts: Redefined Chart Library Built with React and D3,” 2023.
[15] W. McKinney, Python for Data Analysis, 3rd ed. O’Reilly Media, 2022.
[16] Microsoft, “TypeScript Handbook,” 2024.
[17] W3C, “Web Content Accessibility Guidelines (WCAG) 2.2,” 2023.
[18] Akamai Technologies, “State of Online Retail Performance,” 2023.