www.ijarp.com ISSN 2456-9992 Page: 01-13

International Journal Advanced Research Publications

Research Article

Volume: 02
Issue: 05

CRYPTO DASHBOARD: DATA ANALYTICS AND VISUALIZATION FOR CRYPTOCURRENCY MARKETS

*[1]Vishal Kakade, [2]Prof. Hemangi Patil

[1]Dept. of AI & Data Science Ajeenkya DY Patil School of Engineering Pune, India.

[2]Dept. of AI & Data Science Ajeenkya DY Patil School of Engineering Pune, India.

Article Received: 17 April 2026, Article Revised: 07 May 2026, Published on: 27 May 2026

***Corresponding Author: Vishal Kakade**

Dept. of AI & Data Science Ajeenkya DY Patil School of Engineering Pune, India.

DOI: https://doi-doi.org/101555/ijarp.4716

ABSTRACT

Cryptocurrency markets generate continuous high- volume, high-velocity data streams, including real-time prices, trading volumes, and market capitalization for thousands of assets. At the same time, many retail investors and students still rely on fragmented tools such as static price-trackers, mobile applications, and ad-hoc spreadsheets, which makes it difficult to maintain a consolidated view of holdings, iden- tify trends, and take timely decisions. This paper presents Crypto Dashboard, a full-stack web application that applies data analytics and visualization techniques to live cryptocurrency data using a MERN-style technology stack comprising React, Vite, Tailwind CSS, Chart.js, Node.js/Express, and MongoDB. The system integrates real-time price tracking with 30-second auto-refresh, interactive charts, portfolio profit/loss computation, watchlists, and configurable price alerts. Security-aware backend engineering is incorporated through JSON Web Token (JWT)- based authentication, bcrypt password hashing, input validation, rate limiting, Content Security Policy (CSP) headers, and cross- site scripting (XSS) protection, making the platform suitable for deployment in adversarial web environments. Experimental evaluation shows that the dashboards effectively summarize complex market behaviour and portfolio status into intuitive visual interfaces, supporting faster decision-making compared with manual tracking workflows and serving as a reusable template for educational projects on data analytics and web security.

INDEX TERMS: Cryptocurrency, Dashboard, Data analytics, MERN stack, React, Node.js, MongoDB, JWT, Web security, Data visualization.

I. INTRODUCTION

Cryptocurrencies such as Bitcoin, Ethereum, and numer- ous altcoins have evolved into major investment instruments traded around the clock on global exchanges. Their prices are characterized by high volatility and are influenced by market microstructure, macroeconomic announcements, reg- ulatory developments, and social sentiment. Consequently, both professional and retail participants require continuous monitoring and analysis of price and volume time series to make informed decisions.

In practice, individual traders and students frequently rely on a combination of browser tabs, mobile applications, and spreadsheet-based workflows to track holdings. These frag- mented tools lead to duplicated effort, inconsistent data, and limited analytical depth, particularly when managing portfolios that contain dozens of coins across different ex- changes. Business-intelligence and dashboarding techniques have proven effective in other domains for converting large heterogeneous datasets into actionable, visual insights. Ap- plying similar principles to cryptocurrency data can help users reason about price movements, volume spikes, market dominance, and portfolio performance in a more structured way.

Existing academic prototypes often focus on either pre- dictive models (for example, price forecasting or sentiment analysis) or front-end demonstrations with minimal attention to authentication, authorization, and production-grade backend design. Publicly available dashboards tend to fall into three categories: (i) simple price tickers with minimal interactivity,

(ii) exchange-integrated trading terminals optimized for order execution, and (iii) research prototypes centred on a single analytical pipeline. Simple trackers rarely provide portfolio- aware analytics, while trading terminals are typically closed platforms that do not expose their internal architecture or support educational experimentation.

This work addresses these gaps by designing and im- plementing *Crypto Dashboard*, a web- based cryptocurrency analytics and visualization system built with the MERN stack. The application demonstrates how raw market data can be transformed into interactive dashboards and key performance indicators (KPIs) that assist users in understanding trends, managing risk, and evaluating investment decisions in an academic environment.

The remainder of this paper is structured as follows. Sec- tion II defines the problem statement. Section III explains the importance and practical relevance of the system. Section IV outlines the objectives. Section V presents a literature survey of related systems. Section VI describes the system architec- ture and mathematical formulation. Section VII discusses

the development methodology and implementation details. Sec- tion VIII presents results and analysis. Section IX analyses the security posture. Section X concludes the paper and Section XI highlights future scope.

II. Problem Statement

Although numerous cryptocurrency price-trackers and port- folio tools exist, most either provide only market-wide statis- tics or remain closed commercial platforms. There is a lack of open, educational systems that combine real-time data in- gestion, portfolio-aware analytics, and security-aware backend engineering in a single stack. Users must often aggregate in- formation manually across multiple websites and applications, which is error-prone and time-consuming.

The problem addressed in this paper can be stated as follows: *Design and implement a secure, full-stack cryp- tocurrency analytics dashboard that provides real-time price monitoring, portfolio management, watchlists, and alerting features, while following best practices in web application security and software engineering*. The system should serve as both a practical tool for individual users and a pedagogical artefact for courses in data analytics, web technology, and cyber security.

III. Importance of the System

The proposed system plays a significant role in transforming raw cryptocurrency data into actionable intelligence.

First, line, bar, area, and pie charts enable users to interpret complex price and portfolio behaviour through visual repre- sentations such as trend lines, asset allocation plots, and top-gainer views. Instead of scanning raw numeric tables, users can quickly identify which coins are driving performance.

Second, accurate portfolio analytics are supported through server-side logic and client-side calculations that combine user holdings with live market prices. Profit/loss values for each asset and the overall portfolio are computed consistently and updated automatically at each refresh interval. The dashboard exposes both aggregate indicators, such as total portfolio value and daily change, and coin-level indicators, such as percentage contribution to profit and loss.

Third, auto-refresh every 30 seconds ensures that prices and derived metrics remain up to date, enabling faster reactions to market movements than static reports or daily snapshots. Users can detect sudden price changes or potential breakout patterns more effectively, which

is particularly important in markets that trade 24/7.

Fourth, security and trust are central design goals. Features such as JWT-based authentication, hashed passwords, rate limiting, validation, CSP headers, and XSS protection help to protect user data and APIs from common web attacks. This makes the dashboard suitable for use in academic labs and personal projects that are exposed to the public internet.

Finally, the project offers hands-on experience in full- stack development, REST API design, data visualization, and security, which are critical skills for students of artificial intelligence and data science. The architecture is extensible, and can be integrated with machine learning models and streaming pipelines for advanced analytics and research.

IV. Objectives

The main objectives of the Crypto Dashboard project are as follows:

- Develop a full-stack web application for real-time cryp- tocurrency data analytics using the MERN stack (React, Express, MongoDB, Node.js).
- Design interactive dashboards that visualize market trends, asset distribution, and portfolio performance using Chart.js and Tailwind CSS.
- Implement portfolio management features, including adding and removing coins, tracking quantities, and computing profit/loss for each asset and for the overall portfolio.
- Provide watchlist and price alert functionality to support user-defined tracking of selected assets.
- Ensure secure access to user data using JWT authentica- tion, password hashing, input validation, and rate limiting.
- Integrate monitoring and logging (e.g., Sentry) to support debugging, error analysis, and performance monitoring.
- Evaluate the system qualitatively in terms of usability, responsiveness, and completeness of analytics features when compared with common crypto tracker applications.

V. Literature Survey

A growing body of research supports the use of dashboards and analytics tools in finance and cryptocurrency.

Kua [1] developed a web application for cryptocurrency portfolio management using a React-based MERN architec- ture, demonstrating basic CRUD operations on holdings and simple visualizations. The work confirms the feasibility of building portfolio dashboards with

modern JavaScript frame- works but provides limited discussion of security controls such as rate limiting or CSP configuration.

Tran [2] proposed a streaming real-time pipeline for cryp- tocurrency price and news tracking that ingests market and textual data, processes them through a data engineering stack, and exposes dashboards for monitoring price and sentiment trends. While this approach emphasizes streaming infrastruc- ture, it does not support user-specific portfolios or configurable alerts.

Several studies explore cryptocurrency price prediction us- ing forecasting and sentiment analysis. Alharbi et al. [3] com- bine time-series models with sentiment derived from social media to demonstrate the influence of public opinion on price volatility. Other works employ deep learning architectures for price prediction but remain focused on model performance rather than full-stack application design.

Open-source crypto tracker and portfolio projects built with React or Streamlit illustrate basic UI patterns and API integra- tion, often using public APIs such as CoinGecko. However, these projects frequently omit comprehensive authentication and session management, and rarely document their security posture in detail. Beyond cryptocurrency, dashboard design research highlights principles of simplicity, clarity, and ap- propriate visual encodings, which guide layout and charting decisions in this work.

In contrast to the above systems, Crypto Dashboard aims to combine (i) a MERN-based portfolio and watchlist application,

(ii) real-time analytics dashboards, and (iii) explicit security mechanisms such as JWT, bcrypt, Helmet-based CSP, and rate limiting into a cohesive platform suitable both for end users and for teaching.

VI. System Architecture and Mathematical Formulation

The system follows a layered client–server architecture with clear separation of concerns. The main components are the React frontend, Node.js/Express backend, MongoDB database, external cryptocurrency data API, and monitoring tools.

A. High-Level Architecture Diagram

To ensure that diagrams fit cleanly within the IEEE two- column layout, the architecture is drawn as a double-column figure with scaling, as shown in Fig. 1.

B. Data Flow Diagram

Fig. 2 presents the main logical data flow from the user's action to the final JSON response

rendered into charts and tables.

C. *Mathematical Formulation*

Let n denote the number of distinct assets in a user's portfolio. For each asset i, let h_i be the quantity held, b_i the average buy price, and $P_i(t)$ the live market price at time t.

The total portfolio value $V(t)$ is

$$V(t) = \sum_{i=1}^{n} h_i P_i(t). \quad (1)$$

The unrealized profit or loss $\Delta(t)$ is

$$\Delta(t) = \sum_{i=1}^{n} h_i \, P_i(t) - b_i \, . \quad (2)$$

For each asset, the percentage contribution to portfolio value is

$h P (t)$

A. *Requirement Analysis*

Requirements were elicited by observing typical activities of student investors, such as checking prices multiple times per day, manually computing profit/loss, and maintaining ad-hoc watchlists in note-taking applications. Functional require- ments include user authentication, portfolio CRUD operations, watchlist management, alert creation, and multi-chart dash- boards. Non-functional requirements include security, respon- siveness, scalability, and ease of deployment.

B. *Frontend Design and Implementation*

Low-fidelity wireframes were created to decide layout, navigation, and grouping of KPIs. React components were then implemented to represent reusable UI elements, including nav-igation bars, dashboard cards, chart containers, forms for login and registration, and modals for adding or editing portfolio entries. Tailwind CSS was used to ensure a consistent design system with responsive behaviour across desktop and mobile devices. React Router manages navigation between pages such as Dashboard, Market Analytics, Portfolio, Watchlist, Login, and Signup.

State management is handled using React context hooks for authentication, portfolio data, theme selection, and alert state. Chart.js is integrated through wrapper components that

abstract away configuration details and enforce consistent colour palettes and tooltips across line, bar, and pie charts.

C. Backend and Database Design

The backend exposes routes grouped by responsibility (for example, authentication, user, portfolio, alerts). Each route is mapped to controller functions that implement business logic and interact with MongoDB via Mongoose models. The database schema defines user records (email, username, hashed password), portfolio entries (user reference, coin identifier, quantity, purchase price), and alerts (user reference, coin identifier, target price, condition). Indexes on user identifiers and coin identifiers improve query performance for frequent lookups.

The backend includes a service layer that encapsulates calls

$C_i(t) =$

$i \quad i$

$V(t)$

$\times 100\%.$ (3)

to the external crypto API and caches responses briefly in memory to reduce redundant outbound requests. This design

Price alerts are defined by a target price τ_i and a condition

$c_i \in \{\text{above, below}\}$. An alert for asset i at time t is

,

,1, if $c_i =$ above and $P_i(t) \geq \tau_i$,

decouples controller logic from third-party dependencies and simplifies future migration to a dedicated caching tier or message-queue-based streaming architecture.

D. Security Implementation

$A_i(t) =$, 1, if $c_i =$ below and $P_i(t) \leq \tau_i$,

0, otherwise.

(4) Security features were introduced early in the development cycle. Passwords are hashed using bcrypt before storage.

These expressions are evaluated whenever new prices are fetched and are used to compute KPIs and trigger alert notifications.

VII. Methodology and Implementation

The development methodology follows a structured software-engineering and data-analytics lifecycle composed of requirement analysis, design, implementation, testing, and evaluation.

JWT tokens carry user identifiers; short-lived access tokens and longer-lived refresh tokens are used to manage sessions. Express-validator checks request bodies and parameters for correct types and required fields. Express-rate-limit enforces thresholds such as a maximum number of authentication requests per time window. Helmet sets HTTP headers such as HSTS, X-Content-Type-Options, and X-Frame-Options to reduce attack surface. Sanitization tools help mitigate XSS by cleaning user-generated content before rendering.

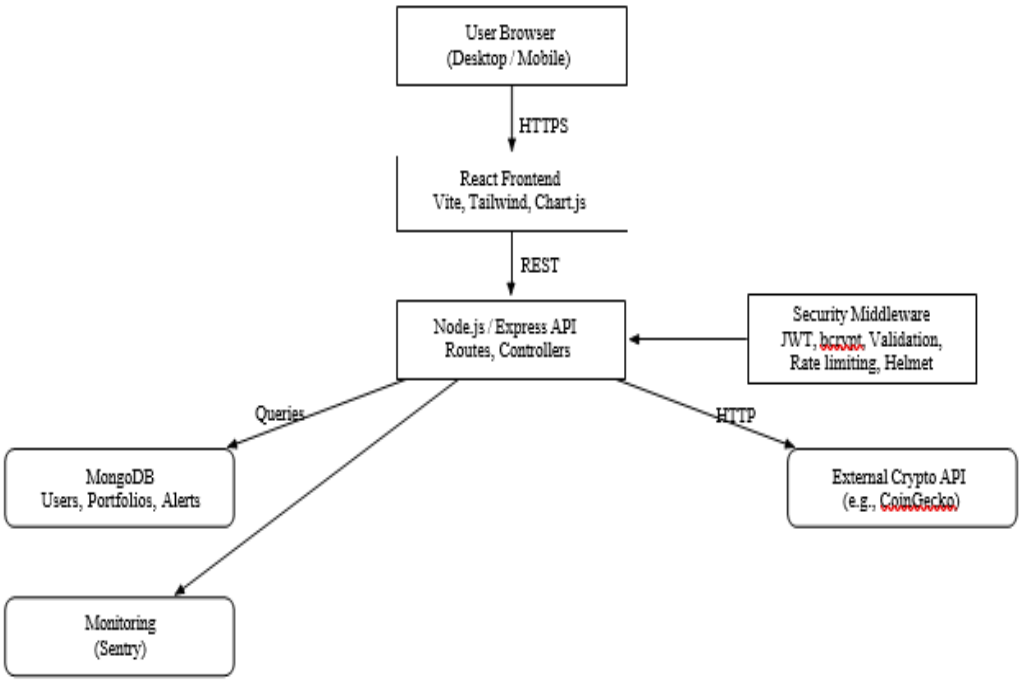

Fig. 1. High-level architecture of the Crypto Dashboard system.

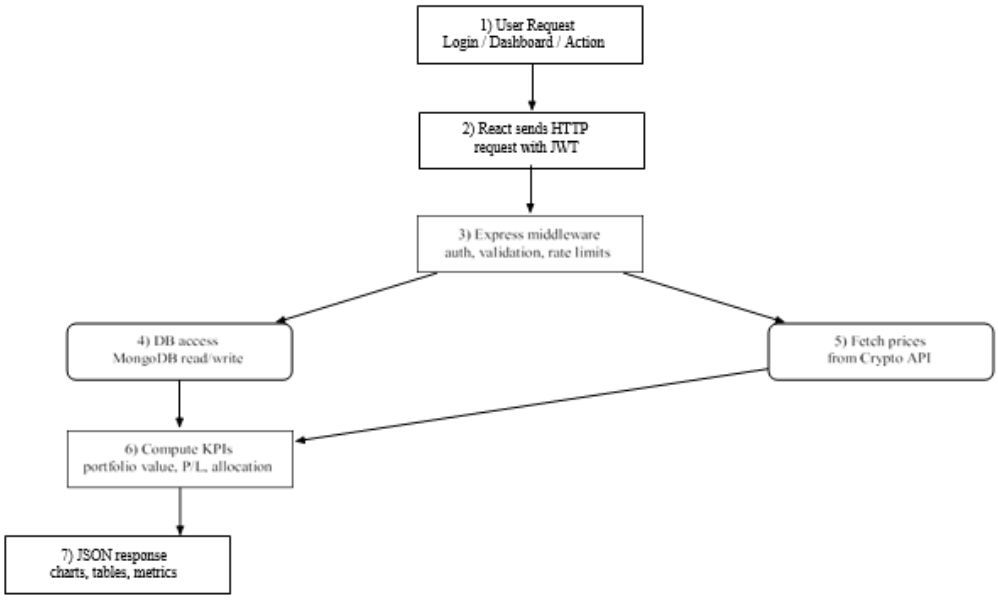

Fig. 2. Simplified data flow in Crypto Dashboard from user request to JSON response.

E. Testing and Continuous Integration

Unit tests for core React components and utility functions were written using a lightweight testing framework. On the backend, syntax checks and route-level smoke tests verify that endpoints load without errors. Git-based workflows and continuous integration are used to run linting, tests, and builds on every push, helping to detect regressions early. Manual exploratory testing validates end-to-end user journeys, including sign up, login, portfolio management, watchlist creation, and alert triggering when prices cross thresholds.

VIII. Results and Analysis

The implemented system satisfies the core functional re- quirements and demonstrates the feasibility of combining real- time data ingestion, analytics, and security in a MERN-based dashboard.

A. Functional Behaviour

Real-time price updates occur at approximately 30-second intervals, maintaining fresh data for all tracked assets. Dash- boards provide multiple interactive visualizations for prices, asset breakdown, and other KPIs, which help users interpret complex data patterns. Portfolio screens allow users to add coins with quantity and purchase price, and automatically compute current value and profit/loss using live prices. Watch- list and alert features streamline monitoring of selected assets and highlight potential trading opportunities without requiring manual refresh.

B. Qualitative Usability

Informal feedback from student users suggests that the dash- board is intuitive to navigate and easier to use than manually updating spreadsheets. Users particularly value the ability to see portfolio-level profit/loss and allocation charts on a single screen, rather than switching between multiple applications. The consistent visual design and responsive layout support use on both desktop and laptop devices.

C. Performance Considerations

During exploratory testing on a mid-range laptop, page navigation and chart updates remained fluid with portfolios of up to several dozen assets. API responses from the external data provider were typically returned within a few hundred milliseconds, and application latency was dominated by net- work delays rather than client-side rendering. However, no formal load testing was conducted, and scaling characteristics under high concurrency remain an open question.

D. Comparison with Baseline Tools

Compared with simple price-only trackers and some open- source portfolio tools, Crypto Dashboard offers a more in- tegrated solution by combining real-time data, personalized analytics, and documented security controls in one stack. Base- line tools typically lack comprehensive security hardening, rarely support user-defined alerts, and do not always maintain persistent portfolios across sessions. The proposed system thus provides both functional and educational advantages.

E. Limitations

The current evaluation is mainly qualitative and based on manual testing. No large-scale load testing or formal user study has been conducted to quantify performance metrics such as maximum concurrent users, request latency under stress, or user satisfaction scores. Additionally, the analytics layer remains descriptive; predictive models and advanced risk metrics are not yet integrated.

IX. Security Analysis

Given that cryptocurrency applications are frequent targets for attacks, the security posture of the system is a critical aspect of the design.

First, the authentication layer uses JWTs with short lifetimes and refresh tokens, which reduces the impact of token leakage while avoiding the need to maintain server-side sessions.

Tokens are stored in HTTP-only cookies to mitigate access from client-side scripts.

Second, rate limiting and validation reduce the risk of brute-force and injection attacks. Authentication endpoints are restricted to a small number of attempts per time window, and request payloads are validated for type, length, and format before further processing.

Third, Helmet-based security headers and CSP rules limit the sources from which scripts and styles can be loaded, making common XSS vectors more difficult to exploit. Com- bined with DOM sanitization in the frontend, this substantially reduces the risk of rendering untrusted content.

Finally, logging and monitoring via Sentry provide visibil- ity into runtime errors and unexpected behaviour, which is essential for detecting misconfigurations and potential abuse in a timely manner. Formal penetration testing remains future work, but the architectural groundwork for secure deployment is in place.

X. Conclusion

This paper presented the design and implementation of Crypto Dashboard, a data analytics and visualization system for cryptocurrency markets using the MERN stack. The sys- tem integrates real-time price ingestion, portfolio tracking, watchlists, alerts, and security-aware backend services into a cohesive platform. The dashboards transform complex, rapidly changing market data into interpretable visual representations that help users understand trends and portfolio behaviour more easily than with manual or fragmented tools.

From an educational perspective, the project demonstrates how modern web technologies, database systems, and security mechanisms can be combined to build practical financial analytics applications. The architecture can also serve as a template for projects in other domains where real-time data, dashboards, and secure user management are required.

XI. Future Scope

Several enhancements can be explored in future work:

- **Predictive Analytics:** integrate time-series models and sentiment analysis modules to forecast price movements and risk, extending the dashboard from descriptive to predictive analytics.
- **Streaming Architecture:** migrate from periodic polling to WebSocket-based streaming or message-queue pipelines for lower-latency updates and improved scalability.
- **Advanced Risk Metrics:** add volatility estimates, Sharpe ratio approximations, drawdown analysis, and correlation heatmaps for multi-asset portfolios.

- **Mobile and PWA Enhancements:** extend progressive web app capabilities for offline caching, push notifica- tions for alerts, and improved mobile user experience.
- **Exchange Integration:** connect to exchange APIs with appropriate security measures to fetch balances or execute simulated trades for educational labs.
- **User Studies and Load Testing:** conduct structured user evaluations and performance benchmarks to quantify usability, comprehension of visualizations, and system behaviour under high concurrency.

REFERENCES

1. A. Kua, "Development of Web Application for Cryptocurrency Portfolio Management in React JS Environment," Bachelor Thesis, 2019.
2. T. Tran, "Exploring Streaming Real-time Pipeline with Cryptocurrency Price and News Tracking System," Bachelor Thesis, Haaga-Helia Uni- versity of Applied Sciences, 2024.
3. M. Alharbi, M. B. Y. Alkhaldi, and A. I. Aljasser, "Cryptocurrency Price Prediction using Forecasting and Sentiment Analysis," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 10, pp. 780–788, 2022.
4. Intrinio, "Building Analytics Dashboards with Real-time Financial Data," technical whitepaper, 2025.
5. K. Mahale, "Data Analytics and Visualization using Power BI," aca- demic project report, Ajeenkya DY Patil School of Engineering.
6. P. Author and Q. Author, "Cryptocurrency Dashboard: Interactive Visu- alization of Market Data," *International Journal of Advanced Research in Science and Technology*, vol. X, no. Y, pp. 1–5, 2025.
7. H. Golchha, "Crypto-Tracker Application Using React-JS," in *Proc. National Conf. on Computing*, 2025.
8. Bitquery, "Crypto Dashboard Project: React-based Cryptocurrency Portfolio Dashboard Using Streaming API," online tutorial, accessed May 2026. [Online]. Available: https://docs.bitquery.io/docs/usecases/ crypto-dashboard/
9. CoinGecko, "Crypto Data API: Most Comprehensive & Reliable Cryp- tocurrency Market Data," online documentation, accessed May 2026. [Online]. Available: https://www.coingecko.com/en/api
10. R. Engineer, "ReactJS Real Time Crypto Dashboard Tutorial," DEV Community article, Feb. 2021. [Online]. Available: https://dev.to/ renaissanceengineer/reactjs-real-time-

crypto-dashboard-tutorial-55m0

11. A. A. Kumar and D. T. L., "Security Measures Implemented in RESTful API Development," *Open Access Research Journal of Engineering and Technology*, vol. 7, no. 1, pp. 105–112, 2024.
12. A. Nandanwar, "Security Best Practices in Node.js: Implementing JWT and MongoDB User Schema," Medium article, May 2024. [Online]. Available: https://medium.com/@ayushnandanwar003/security-best-practices-in-node-js-implementing-jwt-and-mongodb-user-schema-2b5e34da394c
13. Black Label, "How to Create Stunning Financial Dashboard Design,"blog article, Aug. 2025. [Online]. Available: https://www.blacklabel.net/blog/business-insights/how-to-create-stunning-financial-dashboard-design/
14. Wildnet Edge, "Fintech UX Design: Best Practices for Financial Dashboards," blog article, Aug.2025. [Online]. Available: https://www.wildnetedge.com/blogs/fintech-ux-design-best-practices-for-financial-dashboards
15. R. Sahu, A. Singh, S. Yadav, A. Pandey, and U. Dwivedi, "Crypto Tracker using MERN Stack," *International Journal of Innovative Re- search in Technology*, vol. 12, no. 1, pp. 392–394, May 2025.
16. Binance Research, "The Best Cryptocurrency Portfolio Track- ers of 2024," online article, Mar. 2024. [Online]. Available: https://www.binance.com/
17. OWASP Foundation, "API2:2023 Broken Authentication," in *OWASP API Security Top 10*, 2023. [Online]. Available: https://owasp.org/
18. CoinMarketCap, "CoinMarketCap Pro API Documentation," online documentation, accessed May 2026. [Online]. Available: https://coinmarketcap.com/api/
19. R. Konapure, "Cryptocurrency Price and Twitter Sentiment Analysis," GitHub repository, 2021. [Online]. Available: https://github.com/rishikonapure/Cryptocurrency-Sentiment-Analysis
20. CoinGecko Research, "2023 Annual Crypto Industry Report," technical report, Jan. 2024. [Online]. Available: https://www.coingecko.com/