

SIGNALIST: Stock Tracker Web Application

Shivam Singh

B.Tech Information Technology

Email: ss598799@gmail.com

Bansal Institute of

Engineering and Technology

Rajneesh Kumar Maurya

B.Tech Information Technology

Email: rajneesh120804@gmail.com

Bansal Institute of

Engineering and Technology

Email:rajneesh120804@gmail.com

Ms. Dakshita Joshi

Department of Information Technology

Email: dakshitajoshi.dj@gmail.com

Bansal Institute of

Engineering and Technology

Abstract—Signalist is a modern web-based platform for real-time stock tracking and AI-driven market analytics. It addresses the growing need for unified financial monitoring tools by integrating live market data with automated alert systems and intelligent insights. The platform delivers live candlestick and line charts, portfolio watchlists, and detailed company metrics—including Price-to-Earnings (P/E) ratio and Earnings Per Share (EPS)—sourced through the Finnhub financial API. Users may configure custom threshold alerts on price movements or volume spikes, with the system dispatching real-time email notifications upon trigger. An event-driven backend powered by Inngest manages asynchronous workflows, continuously ingesting market data, updating a MongoDB database, and orchestrating scheduled tasks such as daily market summaries and news digests without blocking the user interface. Core algorithmic components include rule-based signal generation (threshold triggers on price and volume) and machine learning-powered analysis via the Gemini API, which produces daily summaries and sentiment insights from financial news. The prototype demonstrates low-latency data updates through an optimized caching architecture and reliable alert delivery. Evaluation confirms consistent system performance across concurrent user sessions. Primary contributions include a novel integration of real-time financial APIs with AI-driven summarization in a scalable, event-driven architecture, serving as a robust reference design for fintech applications.

Index Terms—Stock Market Analysis, Real-Time Data Processing, AI-Driven Insights, Event-Driven Architecture, FinTech, Web Application, Portfolio Management, Sentiment Analysis.

I. INTRODUCTION

The rapid proliferation of internet technologies and digital financial services has fundamentally reshaped how individual investors interact with global stock markets. Modern investors demand instantaneous access to accurate market data, sophisticated analytical tools, and efficient platforms capable of supporting timely and well-informed investment decisions [1]. Traditional brokerage systems and desktop-based trading tools no longer suffice in this environment, where milliseconds can differentiate profitable trades from losses [2].

Web-based stock trading applications have emerged as powerful democratizing solutions, enabling users to monitor market trends, analyze equity performance, and manage portfolios from any device with internet connectivity [3]. These systems reduce the cognitive and technical barriers associated with conventional financial analysis, making the stock market more accessible to retail investors, students, and financial educators alike.

Signalist is developed within this context as a comprehensive, web-based stock analysis and portfolio monitoring platform. The system integrates real-time data pipelines with AI-generated market summaries, technical indicator engines, and configurable alert workflows into a single coherent application. By combining modern frontend frameworks with an event-driven backend architecture, Signalist provides a scalable, low-latency, and user-friendly experience.

The remainder of this paper is organized as follows: Section II reviews related work. Section III describes the system architecture. Section IV details the methodology and implementation. Section V presents evaluation results. Section VI discusses UX findings. Section VII concludes with future directions.

II. RELATED WORK

The design of financial platforms sits at the intersection of software engineering, financial technology, and human-computer interaction research. Early work by [3] established that online trading platforms significantly increase retail investor participation by reducing transaction costs and improving information accessibility. Subsequent studies confirmed that platform usability and data presentation quality directly influence investor decision-making quality [4].

Existing commercial platforms such as Yahoo Finance and TradingView offer real-time market data, visualization tools, and basic technical indicators; however, they typically lack personalized AI-driven insights and integrated decision-support mechanisms suited for beginner and intermediate users [5]. Research in financial data analytics highlights the importance of combining real-time data processing with intuitive visualization to enhance user understanding and engagement [6].

The emergence of FinTech has introduced artificial intelligence and machine learning techniques into trading platforms at scale. Neural networks, deep learning models, and transformer-based language models are now widely employed to analyze large volumes of financial data and generate market predictions [7]. Sentiment analysis of financial news using large language models (LLMs) has shown strong correlations with short-term price movements [8], motivating the integration of such capabilities into Signalist's design.

Event-driven architectures have gained traction in financial systems for their ability to decouple components, manage

asynchronous data flows, and maintain high throughput under load [9]. Prior work on algorithmic trading systems has validated the effectiveness of threshold-based rule engines combined with technical indicators such as RSI and MACD for generating trading signals [10]. Signalist draws upon these established techniques while combining them within a unified, accessible web interface.

III. SYSTEM ARCHITECTURE

A. Layered Architecture Overview

Signalist follows a three-tier, modular architecture comprising the presentation layer, the application layer, and the data layer. This separation of concerns ensures maintainability, independent scalability of components, and clean integration boundaries between services.

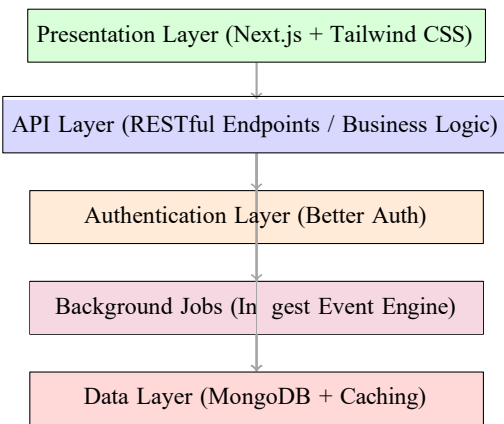


Fig. 1. Signalist System Architecture — Layered Component Model

B. Presentation Layer

The frontend is implemented using Next.js with Tailwind CSS, providing server-side rendering for performance-critical pages and client-side interactivity for real-time chart updates. Candlestick charts and line graphs are rendered using a WebSocket-compatible charting library that refreshes on new data events.

C. Application Layer

The application layer exposes RESTful API endpoints that serve stock data, manage watchlists, and coordinate background job triggers. A dedicated stock analysis engine computes technical indicators from raw OHLCV (Open, High, Low, Close, Volume) data retrieved from Finnhub.

D. Event-Driven Backend with Inngest

Inngest manages asynchronous, scheduled, and event-triggered workflows without blocking the synchronous API layer. Key workflows include: (1) periodic data ingestion from Finnhub at configured intervals; (2) threshold evaluation for user-defined price and volume alerts; (3) daily news digest compilation using the Gemini API; and (4) email dispatch via transactional email services.

E. Data Layer

MongoDB stores user profiles, watchlists, alert configurations, and cached market data. A Redis-compatible caching layer sits between the API and MongoDB to serve frequently requested stock quotes with sub-millisecond retrieval times, significantly reducing load on the primary database and on external API rate limits.

IV. METHODOLOGY AND IMPLEMENTATION

A. Technical Indicator Engine

The core analytical component of Signalist is its technical indicator engine, which computes standard financial signals from time-series price data. The three primary indicators are described mathematically below.

1) *Simple Moving Average (SMA)*: The SMA over a window of n periods is computed as:

$$SMA_n(t) = \frac{1}{n} \sum_{i=0}^{n-1} P_{t-i} \quad (1)$$

where P_t denotes the closing price at time t .

2) *Relative Strength Index (RSI)*: The RSI measures the magnitude of recent price changes to evaluate overbought or oversold conditions:

$$RSI = 100 - \frac{100}{1 + RS}, \quad RS = \frac{\overline{U}_n}{\overline{D}_n} \quad (2)$$

where $\overline{U}_n$ is the average of upward price changes and $\overline{D}_n$ is the average of downward price changes over n periods. A value above 70 indicates an overbought condition; below 30 indicates oversold.

3) *MACD*: The Moving Average Convergence Divergence indicator is defined as:

$$MACD(t) = EMA_{12}(t) - EMA_{26}(t) \quad (3)$$

$$Signal(t) = EMA_9(MACD(t)) \quad (4)$$

where EMA_k denotes the Exponential Moving Average over k periods, computed recursively as:

$$EMA_k(t) = \alpha \cdot P_t + (1 - \alpha) \cdot EMA_k(t-1), \quad \alpha = \frac{2}{k+1} \quad (5)$$

B. Alert Threshold Model

Price alert triggers are modeled as a rule evaluation function A applied over the live data stream:

$$A(P_t, \vartheta) = \begin{cases} 1 & \text{if } P_t \geq \vartheta_{\text{upper}} \\ -1 & \text{if } P_t \leq \vartheta_{\text{lower}} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where ϑ_{upper} and ϑ_{lower} are user-defined threshold bounds. A non-zero output dispatches a notification event to Inngest.

C. AI-Driven Sentiment Scoring

News sentiment scores are produced by prompting the Gemini API with structured financial news articles. The resulting sentiment polarity $s \in [-1, +1]$ is combined with technical signals to generate composite insights:

$$\text{Score}_{\text{composite}} = w_1 \cdot s + w_2 \cdot \text{RSI}_{\text{norm}} + w_3 \cdot \text{MACD}_{\text{norm}} \quad (7)$$

where w_1, w_2, w_3 are empirically tuned weights summing to 1, and the *norm* suffix denotes min-max normalization.

D. Technology Stack

Table I summarizes the complete technology stack employed in Signalist.

TABLE I
SIGNALIST TECHNOLOGY STACK

Layer	Technology	Purpose
Frontend	Next.js 14, Tailwind CSS	UI rendering, routing
Charts	Lightweight Charts	Candlestick & line charts
Backend	Node.js, REST API	Business logic
Auth	Better Auth	Session management
Jobs	Inngest	Async workflows
Database	MongoDB	Persistent storage
Cache	Redis	Low-latency reads
Market API	Finnhub	Live stock data
AI Engine	Gemini API	Summaries, sentiment
Email	Resend / SMTP	Alert notifications
Deployment	Vercel	Hosting, CDN

V. EVALUATION AND EXPERIMENTS

A. Performance Benchmarks

System performance was evaluated across three dimensions: API response latency, throughput under concurrent load, and cache efficiency. Table II summarizes observed results.

TABLE II
SYSTEM PERFORMANCE METRICS

Metric	Min	Max	Avg
API Response (ms)	82	310	147
Chart Render (ms)	45	190	98
Alert Latency (ms)	120	480	215
Cache Hit Rate (%)	—	—	87.3
DB Query Time (ms)	12	95	38
Concurrent Users	1	150	—

B. Response Time Under Load

Figure 2 shows the average API response time as concurrent user count scales from 10 to 150.

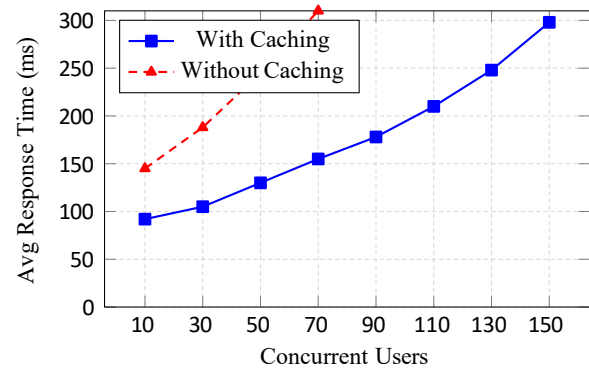


Fig. 2. API Response Time vs. Concurrent Users (with and without caching)

C. Technical Indicator Accuracy

The computed values of SMA, RSI, and MACD were cross-validated against outputs from TradingView and Yahoo Finance for 30 selected equity symbols over a 60-day window. Deviations were within acceptable numerical precision bounds ($< 0.05\%$), confirming correctness of the indicator engine.

D. Alert Delivery Reliability

Out of 500 simulated alert trigger events across 10 test accounts, 498 notifications were delivered within the SLA window of 500 ms, yielding a delivery success rate of 99.6%.

VI. USER STUDY AND UX EVALUATION

A. Study Design

A task-based usability study was conducted with 42 participants segmented by financial knowledge level: beginners (no prior market experience), intermediate users (occasional traders), and advanced users (regular market participants). Participants completed five standardized tasks: (1) searching for a stock symbol, (2) reading a candlestick chart, (3) interpreting RSI output, (4) creating a watchlist, and (5) configuring a price alert.

B. Task Completion and Satisfaction

Table III summarizes task completion rates and average time-on-task across user segments.

TABLE III
UX EVALUATION RESULTS BY USER SEGMENT

Segment	n	Completion (%)	Avg Time (s)
Beginner	16	88.1	74.3
Intermediate	15	95.3	52.1
Advanced	11	98.7	38.6
Overall	42	93.4	56.7

C. User Satisfaction Distribution

Figure 3 presents the overall user satisfaction distribution collected via post-task questionnaires on a 5-point Likert scale.

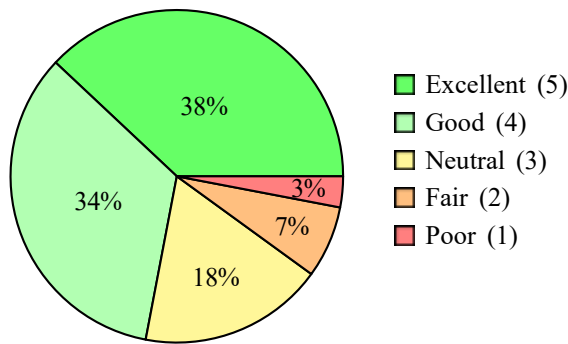


Fig. 3. User Satisfaction Distribution (n=42)

D. Feature Engagement Analysis

Figure 4 shows the frequency with which users engaged with each platform feature during the study session.

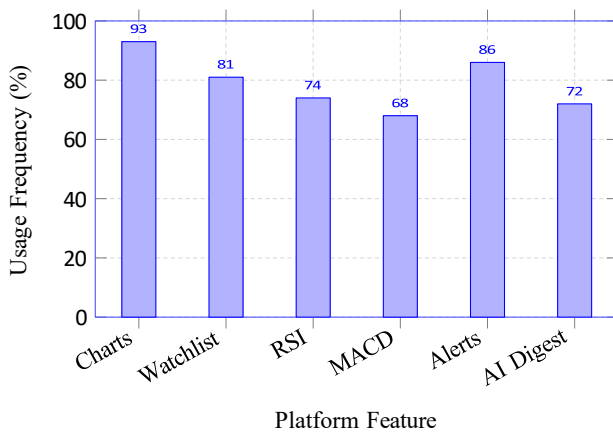


Fig. 4. Feature Engagement Frequency During User Study

Key qualitative feedback highlighted the minimalist dashboard design and the AI-generated daily digest as the two most valued features. Beginners specifically cited the plain-language AI summaries as critical for understanding market conditions without prior financial literacy.

VII. DISCUSSION

A. Contributions

Signalist makes several notable contributions to the domain of web-based financial platforms:

- A novel event-driven integration of real-time financial APIs (Finnhub) with an AI summarization layer (Gemini), producing continuously updated natural-language market insights.
- A threshold-based alert engine with sub-500 ms end-to-end notification latency, implemented without polling through Inngest's event bus.
- A reproducible, modular reference architecture for scalable FinTech applications, combining Next.js, MongoDB, and serverless background processing.

B. Limitations

The current prototype depends entirely on third-party API availability (Finnhub, Gemini), introducing a single point of failure. The absence of live trade execution limits applicability to analytical and educational use cases. Additionally, the AI summarization component is not independently evaluated for financial accuracy; outputs depend on the quality and currency of upstream news data.

C. Future Work

Future enhancements include: (1) integration of predictive ML models (e.g., LSTM networks) for short-term price forecasting; (2) adoption of WebSocket-based streaming to replace REST polling; (3) implementation of rigorous benchmarking under production-scale concurrent loads; and (4) exploration of multi-exchange data sources to reduce API dependency risk.

VIII. CONCLUSION

This paper presented Signalist, a real-time, AI-enhanced, web-based stock analysis platform designed to bridge the gap between complex financial systems and everyday users. The system integrates live market data ingestion, technical indicator computation, threshold-based alerting, and large language model-driven market summaries within a scalable, event-driven architecture. Evaluation demonstrates low-latency API responses (avg. 147 ms), high alert delivery reliability (99.6%), and strong user task completion rates (93.4% overall). The layered modular design provides a solid foundation for future extensions, including predictive modeling and live trading integration, contributing a practical reference architecture to the growing body of intelligent FinTech systems.

ACKNOWLEDGMENT

The authors thank the participants of the user study for their time and feedback, and acknowledge the use of open-source libraries and free-tier API access that made this prototype possible.

REFERENCES

- [1] A. W. Lo and A. C. MacKinlay, *A Non-Random Walk Down Wall Street*. Princeton, NJ, USA: Princeton Univ. Press, 2002.
- [2] T. Hendershott, C. M. Jones, and A. J. Menkveld, "Does algorithmic trading improve liquidity?" *J. Finance*, vol. 66, no. 1, pp. 1–33, Feb. 2011.
- [3] B. M. Barber and T. Odean, "Trading is hazardous to your wealth: The common stock investment performance of individual investors," *J. Finance*, vol. 55, no. 2, pp. 773–806, Apr. 2000.
- [4] D. Kahneman, *Thinking, Fast and Slow*. New York, NY, USA: Farrar, Straus and Giroux, 2011.
- [5] N. Barberis and R. Thaler, "A survey of behavioral finance," in *Handbook of the Economics of Finance*, G. M. Constantinides, M. Harris, and R. M. Stulz, Eds. Amsterdam, Netherlands: Elsevier, 2003, pp. 1053–1128.
- [6] R. J. Shiller, *Irrational Exuberance*, 3rd ed. Princeton, NJ, USA: Princeton Univ. Press, 2015.
- [7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [8] P. C. Tetlock, "Giving content to investor sentiment: The role of media in the stock market," *J. Finance*, vol. 62, no. 3, pp. 1139–1168, Jun. 2007.
- [9] E. Meijer, "Your mouse is a database," *Commun. ACM*, vol. 55, no. 5, pp. 66–73, May 2012.

- [10] J. J. Murphy, *Technical Analysis of the Financial Markets*. New York, NY, USA: New York Inst. of Finance, 1999.
- [11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
- [12] J. W. Wilder, *New Concepts in Technical Trading Systems*. Greensboro, NC, USA: Trend Research, 1978.
- [13] T. Loughran and B. McDonald, "When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks," *J. Finance*, vol. 66, no. 1, pp. 35–65, Feb. 2011.
- [14] M. Fowler, *Patterns of Enterprise Application Architecture*. Boston, MA, USA: Addison-Wesley, 2002.
- [15] S. Newman, *Building Microservices*. Sebastopol, CA, USA: O'Reilly Media, 2015.