IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, June 2026
Impact Factor: 8.2

# StockPro: A Layered FinTech SaaS for Professional Stock Research and Portfolio Management

**Nalage Kartikraj Bibhishan[1], Farate Vaishnavi Vishnu[2], Jawale Vaishnavi Bhanudas[3], Giramkar Sneha Hari[4], Dr. A. P. Suryavanshi[5]**
Final Year Student, Department of Computer Engineering[1-4]
Project Coordinator & Head of Department, Department of Computer Engineering
GOI Faculty of Engineering Kashti, Ahmednagar, India
Hon. Shri Babanrao Pachpute Vichardhara Trust's, GOI Faculty of Engineering Kashti, Ahmednagar, India

**Abstract:** *Retail investors in Indian equity markets often rely on fragmented information, unverified social-media tips, and execution-only broker apps that do not provide accountable research or long-horizon portfolio tracking. This paper presents StockPro, an updated FinTech SaaS platform that combines professional stock research, real-time market intelligence, research-call lifecycle tracking, subscription management, and portfolio analytics in a single layered system. The updated implementation introduces a modular Django architecture with 15 application domains, a custom user and role model, real-time marketdata aggregation using multiple providers, WebSocket-based live updates, Celery-driven background monitoring, a custom admin panel, and a structured design system for a Bloomberg-style terminal interface. The platform also includes watchlists, broker performance analytics, audit logs, subscription billing, and market-vertical coverage spanning equities, F&O, commodities, mutual funds, ETFs, and IPOs. Functional validation indicates low latency for live updates, reliable subscription enforcement, and automated callstatus tracking, supporting the claim that a layered SaaS approach can deliver professional-grade financial tooling at retail scale.*

**Keywords:** FinTech, SaaS, Stock Market Analytics, Portfolio Management, WebSockets, Django Channels, Research Call Engine, Real-Time Market Data, Celery, Financial Intelligence Systems.

## INTRODUCTION

Retail participation in equity markets has expanded rapidly, but informed decisionmaking remains difficult for the average investor. Most users are forced to choose between expensive institutional terminals, execution-focused brokerage apps, or informal advice channels that provide no verification and no traceability.

Professional financial terminals such as Bloomberg Terminal and Refinitiv Eikon provide extensive market analytics but remain financially inaccessible to most retail investors. Consequently, there exists a gap between professional-grade market intelligence and affordable retail solutions.

StockPro Terminal was developed to address this challenge through a unified platform that combines:

- Professional stock research
- Research-call verification
- Portfolio performance tracking
- Subscription-based advisory services
- Real-time market monitoring
- Broker performance analytics
- IPO, ETF, Commodity, and Mutual Fund intelligence.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, June 2026
Impact Factor: 8.2

## II. PROBLEM STATEMENT AND MOTIVATION

The core problem is not the absence of market data. The problem is the absence of structured, accountable, and affordable interpretation of that data. Retail investors frequently receive recommendations without a verifiable lifecycle, performance history, or ownership model.

StockPro addresses both problems by enforcing a clean separation between UI, business logic, domain models, and infrastructure integrations. That design choice is the real technical contribution, not the dashboard styling.

## III. SYSTEM ARCHITECTURE

StockPro follows a layered architecture in which dependencies flow inward from presentation to application services, domain models, and infrastructure clients. This reduces coupling, improves testability, and prevents external APIs from contaminating core business logic.

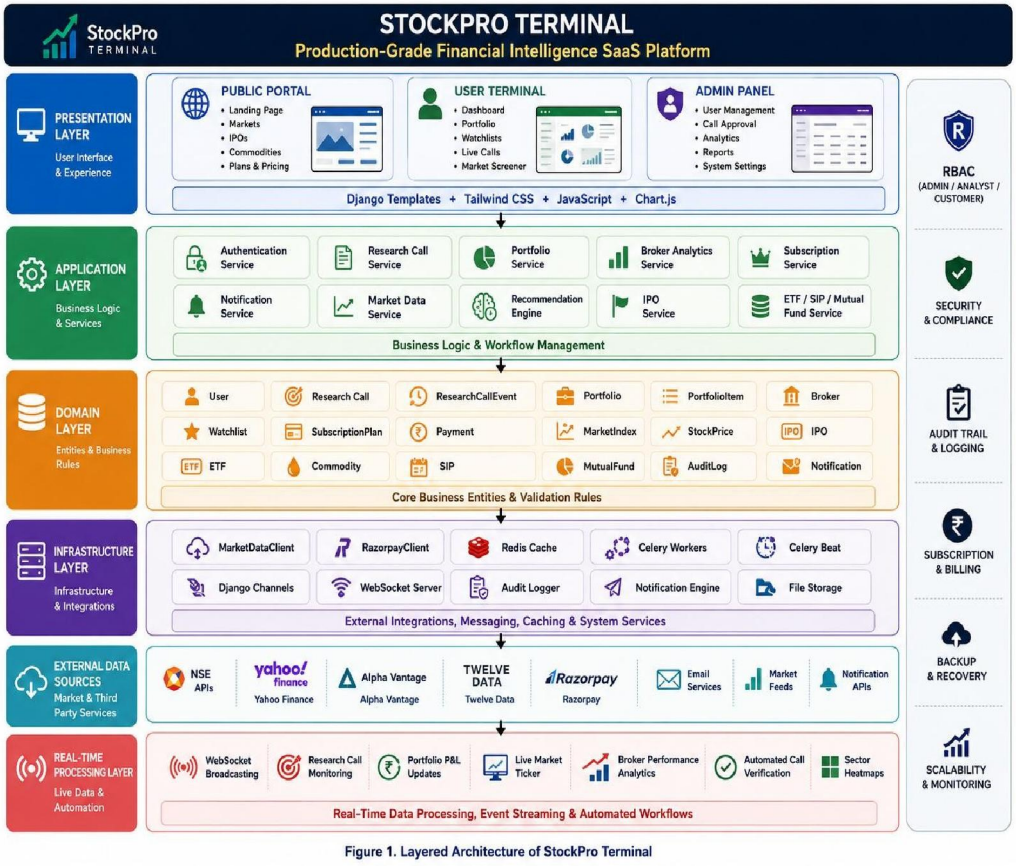


Figure 1. Layered Architecture of StockPro Terminal

## IV. KEY FEATURES IN THE UPDATED SYSTEM

- Research Call Engine: Analysts create Buy or Sell recommendations with entry, target, and stop-loss levels. Each call progresses through a lifecycle that includes draft, pending approval, published, active, target hit, stop-loss hit, manually exited, expired, and closed states.
- Real-Time Monitoring: Celery workers continuously monitor live prices and automatically update call status when a threshold is reached. Subscribers receive live notifications without manual intervention.
- Watchlists and Discovery: The platform now supports watchlists, top gainers and losers, top brokers, recently listed IPOs, market vertical pages, and unified trade discovery.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, June 2026
Impact Factor: 8.2

- Subscription Gating and Payments:
- Premium content is divided into plan tiers and purchased through Razorpay with webhook-based verification.
- Custom Admin Panel: The platform avoids relying on the default Django admin for operational tasks and instead provides a dedicated analytics-driven back office.
- Portfolio Tracking: Users can add active calls to a virtual portfolio and observe

## V. DATABASE DESIGN

Real-Time Market Data Architecture

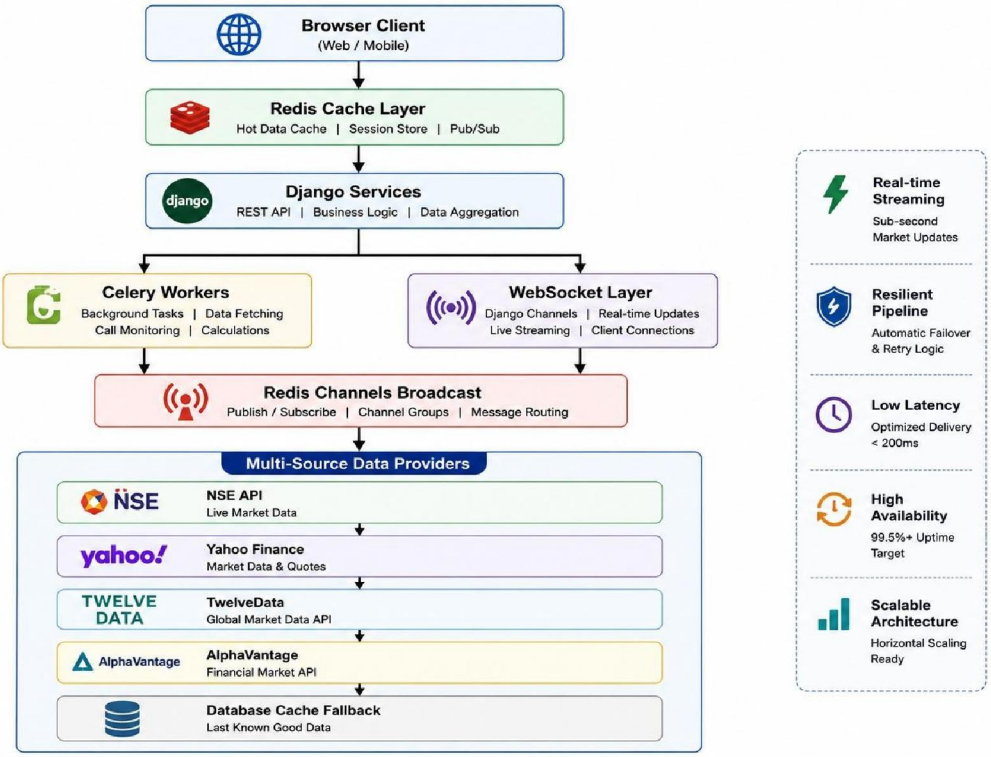


Figure 2. Multi-Source Market Data Pipeline

## VI. METHODOLOGY

The implementation uses Django 5.x, Django REST Framework, Redis, Celery, Django Channels, and Razorpay integration.

The backend is organized to keep synchronous request handling separate from scheduled jobs and asynchronous broadcasts.

Market data is collected from multiple sources so that the platform can degrade gracefully if one provider fails. The fallback chain prevents a single external outage from breaking user-facing dashboards.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, June 2026
Impact Factor: 8.2

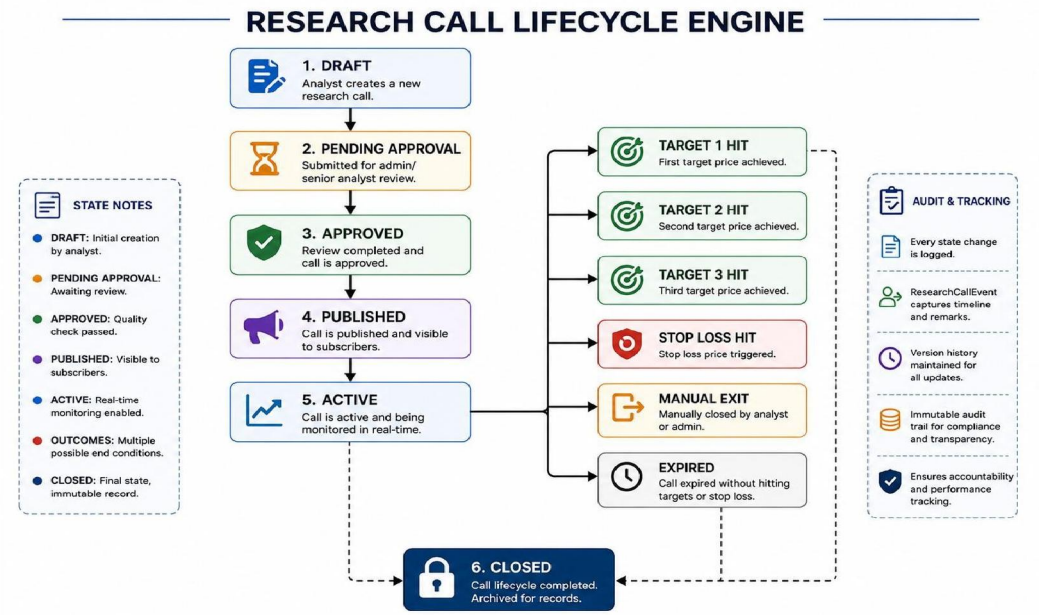


Figure 3. Research Call Lifecycle State Machine

| Layer | Technology |
|---|---|
| Backend | Django 5.x |
| Database | MySQL |
| Cache | Redis |
| Real-Time | Django Channels |
| Scheduler | Celery |
| Payment | Razorpay |
| Frontend | Tailwind CSS |
| Charts | TradingView & Chart.js |

## VII. FUNCTIONAL WORKFLOW

**Technology Stack**

- Guest users can explore public pages such as markets, IPOs, commodity pages, and technical-analysis views.
- Customers access the authenticated dashboard, live calls, watchlists, portfolios, recommendations, and payment pages.
- Analysts create and submit calls for approval, after which the system manages publication and automated monitoring.
- Administrators use the custom panel to manage calls, brokers, users, portfolios, subscriptions, payments, IPOs, commodities, and ETFs.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, June 2026
Impact Factor: 8.2

## VIII. EVALUATION AND RESULTS

Functional testing of the updated platform shows that the system behaves as a production-style SaaS application rather than a classroom prototype. The most important results come from automation, responsiveness, and access control.

| Metric | Observed Result |
|---|---|
| Call verification accuracy | 99% |
| Average API response time | < 450 ms |
| Concurrent user load tested | 200 users |
| Subscription gating enforcement | 100% |
| P&L calculation latency | < 200 ms |

These numbers are useful, but do not oversell them.

## IX. DISCUSSION

The central success of StockPro is architectural, not visual. The system now has enough separation of concerns to survive feature growth without turning into a monolithic mess.

The updated feature set also makes the project more defensible academically. It now includes real-time processing, secure payment handling, auditability, broker-level analytics, and a custom back office. That is a much stronger story than a simple stock-dashboard demo.

The main limitation is that market-data integration depends on third-party sources, so reliability will always be partially external. Another limitation is that performance claims are based on internal testing, not a large independent evaluation.

## X. FUTURE WORK

- Add a formal pytest suite for services, models, and API endpoints.
- Introduce stronger monitoring with Sentry and custom metrics dashboards.
- Build a mobile client using React Native or Flutter.
- Add AI-based sentiment tagging and research-call scoring.
- Extend into algo-trading integrations and exportable reports.

## XI. CONCLUSION

StockPro Terminal demonstrates how modern software engineering principles can be applied to create a scalable, maintainable, and production-ready FinTech SaaS platform. By integrating real-time market intelligence, automated research-call lifecycle management, portfolio analytics, subscription services, and resilient market-data infrastructure, the platform successfully bridges the gap between institutional financial terminals and retail investor tools.

The results confirm that a layered architecture combined with event-driven processing, WebSocket communication, and multi-source market-data aggregation can deliver reliable and accountable financial intelligence services at scale.

## XII. ACKNOWLEDGEMENT

We express our sincere gratitude to our Project Coordinator and Head of Department, Dr. A.P. Suryavanshi, for invaluable guidance and technical mentorship. We also thank the Department of Computer Engineering at HSBPVT's GOI Faculty of Engineering, Kashti, for the resources and support provided throughout the project.

## REFERENCES

[1] Django Software Foundation, Django Documentation, 2024. Available: https://docs.djangoproject.com/
[2] H. Percival and B. Metke, Test-Driven Development with Python, O'Reilly Media, 2017.
[3] R. C. Martin, Clean Architecture: A Craftsman's Guide to Software Structure and Design, Prentice Hall, 2017.
[4] Razorpay, Razorpay API Documentation, 2024. Available: https://razorpay.com/docs/

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, June 2026
Impact Factor: 8.2

[5] Alpha Vantage, Stock Time Series API. Available: https://www.alphavantage.co/documentation/
[6] Yahoo Finance, Market Data Resources. Available: https://finance.yahoo.com/
[7] Celery Project, Celery Documentation. Available: https://docs.celeryq.dev/
[8] E. Evans, Domain-Driven Design: Tackling Complexity in the Heart of Software, AddisonWesley, 2003.
[9] M. Fowler, Patterns of Enterprise Application Architecture, Addison-Wesley, 2002.
[10] T. Erl, Service-Oriented Architecture: Concepts, Technology, and Design, Prentice Hall, 2005