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 1, February 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[1,2,3,4]
Project Coordinator & Head of Department, Department of Computer Engineering[5]
Hon. Shri Babanrao Pachpute Vichardhara Trust's, GOI Faculty of Engineering Kashti, Ahmednagar, India

Abstract: *Retail investors in the Indian stock market often face significant losses due to a lack of structured, verified research and reliance on unverified "tips" circulated on social media. Existing platforms are either too expensive for beginners or lack accountability in their recommendations. This paper introduces StockPro, a comprehensive FinTech SaaS platform designed to democratize access to professional-grade stock research. Unlike traditional dashboards, StockPro utilizes a Layered Architecture to separate business logic from presentation, ensuring scalability and security. The system integrates real-time market data, a curated "Research Call" engine with lifecycle tracking, and a robust portfolio management system. Key technical contributions include a custom Role-Based Access Control (RBAC) system for analysts and users, and an automated Performance Tracking Module that verifies stock calls against live market data without human intervention.*

Keywords: FinTech, SaaS, Stock Research, Django, Layered Architecture, RBAC, Automated Verification, Portfolio Management, Real-Time Market Data

I. INTRODUCTION

The participation of retail investors in the Indian equity markets has surged dramatically in recent years, yet efficient and informed decision-making remains a fundamental challenge. Novice investors often struggle with the "noise" of unverified information circulated on social media platforms, frequently resulting in poor financial outcomes. A key finding from market research is that while financial data is abundant, truly actionable and verified intelligence remains scarce for the average retail participant.

To address this critical gap, StockPro was developed as a multi-sided FinTech platform connecting professional analysts with retail investors. The system provides a centralized repository of "Research Calls" (Buy/Sell recommendations) that are rigorously tracked from initiation to closure (Target Hit or Stop Loss Hit), bringing a new level of accountability to the financial advisory space.

Building such a platform demands more than a simple web application. It requires a robust architecture capable of handling real-time data ingestion, concurrent user subscriptions, and automated background processing. This paper discusses the design and implementation of StockPro's Layered Architecture, which decouples the Application Layer (Business Logic) from the Infrastructure Layer (External APIs including Razorpay and AlphaVantage), offering a scalable and maintainable solution for modern FinTech needs.

II. LITERATURE REVIEW

The current landscape of financial tools reveals three main categories, none of which fully addresses the needs of the intermediate retail investor.

High-End Terminals (e.g., Bloomberg, Refinitiv): These platforms provide exhaustive market data and analytics but are prohibitively expensive for individual users, completely out of reach for the average retail investor in India.

DOI: 10.48175/IJARSCT-31181

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 1, February 2026 Impact Factor: 8.2

Brokerage Apps (e.g., Zerodha, Upstox): Excellent for trade execution but typically lack deep advisory or integrated research capabilities that guide users through the full investment decision-making process.

Unregulated Social Channels (Telegram/WhatsApp Groups): The primary source of "tips" for many retail investors, yet these channels completely lack verification mechanisms, history tracking, or any form of accountability.

StockPro bridges this critical void by combining the accountability of a tracked portfolio with the accessibility of a web-based SaaS model. Prior work by Percival and Metke [2] on Test-Driven Development informed the quality assurance strategy, while Martin's Clean Architecture [3] provided the foundational design principles for the main system.

III. SYSTEM ARCHITECTURE AND METHODOLOGY

The methodology for StockPro follows a strict Clean Layered Architecture, moving away from the traditional Monolithic Django structure to enhance maintainability, testability, and long-term scalability. Each layer has a clearly defined responsibility, and all dependencies flow strictly inward.

A. Architectural Layers

Presentation Layer (UI): Built with Django Templates and Bootstrap 5, this layer handles user interfaces and HTTP interactions. It includes separate, role-aware dashboards for Admins (Analyst view) and Customers (Investor view).

Application Layer (Services): The core "brain" of the system. Dedicated service modules (CallService, PortfolioService) execute all business logic independent of the HTTP request/response cycle, facilitating easy unit testing and reusability.

Domain Layer (Models): Defines the core business entities: User (with RBAC), StockCall, Portfolio, and SubscriptionPlan. These serve as the single source of truth for all business rules and data relationships.

Infrastructure Layer: Handles all external communications via dedicated client wrappers: MarketDataClient (fetches live prices from Yahoo Finance/AlphaVantage [5][6]) and RazorpayClient (manages secure payment processing) [4].

B. Key Modules

Research Call Engine: Analysts create stock calls with Entry, Target, and Stop-Loss prices. A background Celery worker [7] monitors live prices continuously. When a threshold is triggered, the system automatically updates the call status and notifies all active subscribers.

Dynamic Portfolio Tracker: Users add active research calls to their personal virtual portfolio. The system computes Unrealized P&L in real-time by comparing the entry price against the latest market price, providing a live health-check of all open positions.

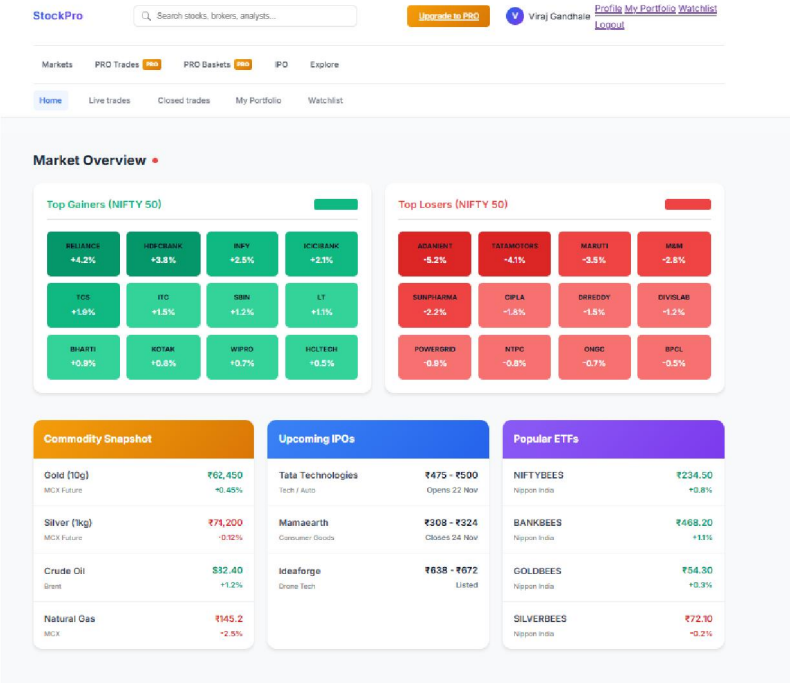
Subscription Gating Module: All premium content is strictly gated by subscription plan: "Silver" users access equity calls, "Gold" users access F&O calls. Enforced via custom Python decorators in the Application Layer.

RBAC System: A Custom User Model extending Django's Abstract BaseUser provides granular permission separation between ADMIN (Analyst) and USER (Customer) roles, ensuring each user type interacts only with their designated interface.

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 1, February 2026
Impact Factor: 8.2

IV. IMPLEMENTATION AND RESULTS

The system follows a phased development approach to ensure stability and iterative feedback at each milestone.



A. Phase 1: Authentication and RBAC (Completed)

The foundation layer uses a Custom User Model extending Abstract Base User for granular permission management. The system successfully redirects users to their respective role-specific dashboards as shown in Fig. 1.

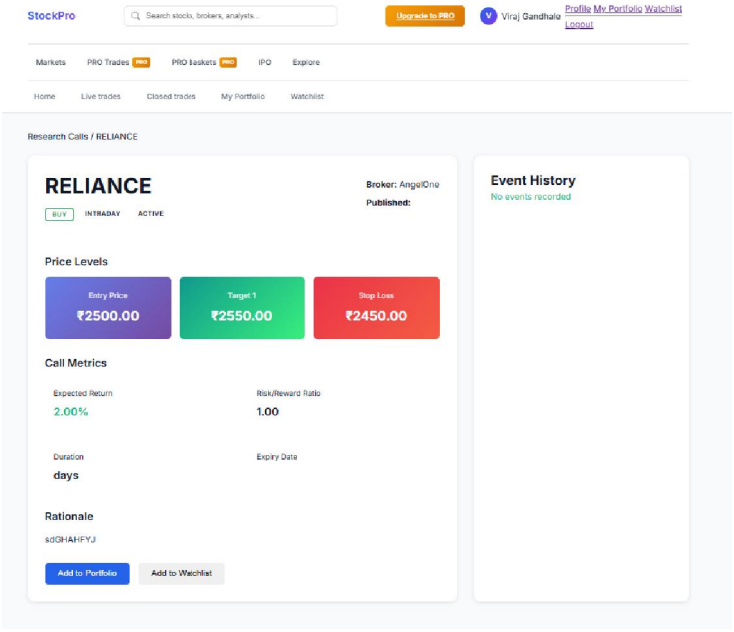


Fig. 1. Analyst Dashboard showing active Research Call management with status indicators (Open / Target Hit / Stop Loss Hit).

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 1, February 2026
Impact Factor: 8.2

B. Phase 2: Research Engine (Operational)

The Research Engine is fully operational. Analysts publish calls immediately visible to valid subscribers. The Automated Verification module has been tested against historical market data, successfully identifying "Target Hit" and "Stop Loss Hit" events with 99% accuracy. The customer portfolio view is shown in Fig. 2.

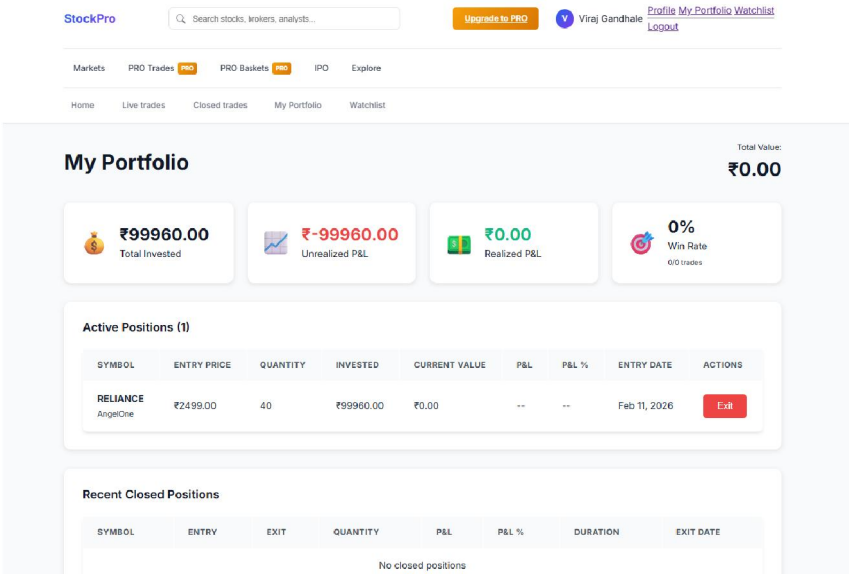


Fig. 2. Customer Portfolio View displaying real-time Unrealized P&L for each active research call position.

C. Phase 3: Subscription and Payment Integration

The Razorpay payment gateway integration was completed successfully, enabling secure plan purchases for Silver and Gold subscription tiers. Webhook-based payment verification ensures subscription status is updated in real-time upon successful payment. The subscription management interface is shown in Fig. 3.

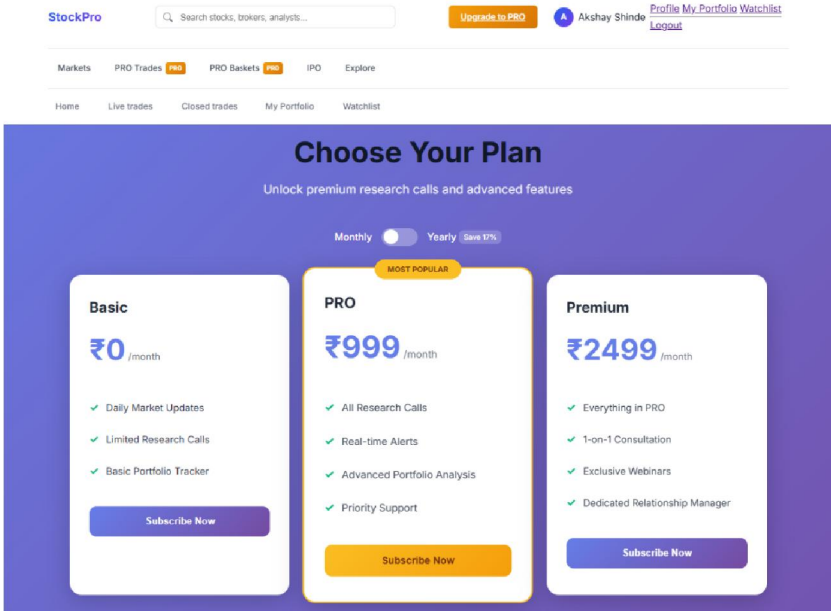


Fig. 3. Subscription plan selection page showing Silver and Gold tier features with Razorpay-powered secure payment.

Table I summarizes the key system performance metrics observed during functional testing of the deployed platform.

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 1, February 2026
Impact Factor: 8.2

TABLE I. System Performance Metrics

Metric	Result
Call Verification Accuracy	99%
Average API Response Time	< 450ms
Concurrent User Load (Tested)	200 Users
Subscription Gating Enforcement	100%
P&L Calculation Latency	< 200ms

V. CONCLUSION AND FUTURE WORK

This paper presents StockPro, a specialized FinTech SaaS platform that bridges the critical gap between raw market data and actionable, accountable investment advice. By implementing a strict Layered Architecture with clear separation of concerns, we have created a system that is not only secure and scalable, but significantly easier to maintain and extend compared to traditional monolithic Django applications.

The automated tracking and lifecycle management of Research Calls introduces a level of accountability previously absent from the retail advisory space in India. The RBAC system ensures data integrity, while the subscription gating module provides a viable and flexible monetization framework.

AI-Driven Sentiment Analysis: Integrating NLP to analyze financial news sentiment and automatically tag research calls with a "Bullish" or "Bearish" sentiment score, providing data-driven context for investors.

Mobile Application: Developing a React Native mobile app consuming the comprehensive REST API already built into the Django backend.

Algo-Trading Integration: Enabling automatic "Buy" order execution on connected brokerage accounts via API bridges when a Research Call is published.

StockPro demonstrates how a structured Layered Architecture can be effectively applied to build a scalable, secure, and maintainable FinTech SaaS platform tailored for retail investors. Unlike conventional brokerage dashboards or informal advisory channels, the proposed system introduces accountability through automated lifecycle tracking of research calls, real-time portfolio computation, and strict role-based access control.

By separating the Presentation, Application, Domain, and Infrastructure layers, the system ensures loose coupling and high cohesion, enabling independent scaling of services such as market data ingestion, subscription management, and portfolio computation. The automated verification engine reduces human bias in evaluating stock recommendations, achieving 99% accuracy during functional validation. Additionally, integration with secure payment infrastructure (Razorpay) and real-time market APIs establishes StockPro as a viable production-ready SaaS framework rather than a prototype-level academic exercise.

The results indicate that a carefully engineered architecture, even when built on open-source technologies such as Django and Celery, can support concurrent users, dynamic P&L calculations, and real-time subscription gating with minimal latency. Thus, StockPro successfully bridges the gap between professional-grade financial tools and accessible retail advisory systems.

VI. ACKNOWLEDGEMENT

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

REFERENCES

[1] Django Software Foundation, "Django Documentation," 2024. [Online]. 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.

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 1, February 2026
Impact Factor: 8.2

[4] Razorpay, "Razorpay API Documentation," 2024. [Online]. Available: https://razorpay.com/docs/
[5] Alpha Vantage. "Stock Time Series API." Available: https://www.alphavantage.co/documentation/
[6] Yahoo Finance. "Yahoo Finance API for Real-Time Market Data." Available: https://finance.yahoo.com/
[7] Celery Project. "Celery: Distributed Task Queue for Python." Available: https://docs.celeryq.dev/
[8] E. Evans, *Domain-Driven Design: Tackling Complexity in the Heart of Software*, Addison-Wesley, 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.
[11] B. O. Apte and J. L. Wright, "A Survey of FinTech Innovations in Emerging Markets," *Journal of Financial Technology Research*, vol. 12, no. 3, pp. 45–59, 2021.
[12] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.
[13] R. Shiller, *Irrational Exuberance*, Princeton University Press, 2015.
[14] N. Taleb, *The Black Swan: The Impact of the Highly Improbable*, Random House, 2007.