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 4, November 2026
Impact Factor: 8.2

Smart AI Trading Chatbot: A Conversational Stock-Market Analysis and Decision-Support System

Bhagwat Rohan Vitthal, Karande Shubham Ashok, Ghule Rutwik Sanjay, Padole Sanket Bhiva
Head of Department: H. B. Jadhav and Guide: P. B. Palwe
Department of Computer Engineering
Adsul's Technical Campus, Chas, Ahilyanagar, Maharashtra, India

Abstract: *Smart AI Trading Chatbot is proposed as a browser-based application for stock search, historical price visualisation, watchlists, and AI-assisted questions. It combines HTML, CSS, and JavaScript with a Python Flask backend, a separate stock-data provider, an AI service, Chart.js, and SQLite. Flask validates market responses and supplies relevant facts to the AI service for plain-language explanation. The analysis layer may calculate returns, moving averages, RSI, MACD, and volatility. Only when data are sufficient and documented rules are met may it show an explained Buy, Hold, or Sell research label with source, timestamp, and risk notes. The AI service does not supply live prices, and the prototype does not execute trades or guarantee returns.*

Keywords: stock market analytics, conversational AI, financial time series, technical indicators, Flask, Chart.js, explainable decision support

I. INTRODUCTION

Stock prices, company details, charts, and explanations are spread across separate services. Beginners may struggle to compare price movements, interpret indicators, and separate reliable market data from speculation. This project brings supported stock search, visualisation, and conversational support into one interface.

Users can view historical prices, maintain a watchlist, and ask questions in natural language. Flask coordinates market-data retrieval and AI explanations, Chart.js renders time series, and SQLite stores selected records. Any Buy, Hold, or Sell label is conditional on documented criteria and adequate data; the prototype does not execute trades.

stores selected records. Any Buy, Hold, or Sell label is conditional on documented criteria and adequate data; the prototype does not execute trades.

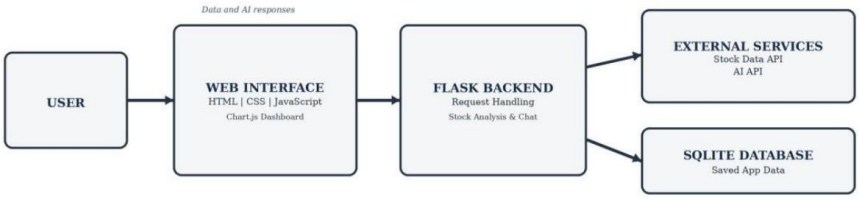

Figure 1. Proposed Architecture of the Smart AI Trading Chatbot

Fig. 1. Proposed Architecture of the Smart AI Trading Chatbot

II. LITERATURE REVIEW

Markowitz [1] linked investment return with risk, while Fama [2] discussed the efficient-market hypothesis and limits of forecasting. Kumbure et al. [3] surveyed machine-learning methods and datasets for stock prediction, while

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 4, November 2026 Impact Factor: 8.2

Olorunnimbe and Viktor [4] reviewed deep-learning practice and backtesting. These studies underline noisy financial time series and the need for sound data, features, and testing. Accordingly, the system separates retrieved facts from AI explanations, shows sources and timestamps, and treats any signal as an uncertain research aid.

III. CASE AND METHODOLOGY

The HTML, CSS, and JavaScript frontend provides the dashboard, stock search, charts, and chatbot. Python Flask validates requests and coordinates a stock-data API and an AI service. Chart.js renders historical prices, while SQLite is planned for watchlists and preferences. API credentials remain on the server; provider coverage, data delay, licensing, and rate limits determine the available data.

For each request, Flask validates the symbol and date range, retrieves historical prices, normalises records, and returns the data source and timestamp. The analysis module may calculate returns, moving averages, RSI, MACD, and rolling volatility.

The AI service receives validated facts and calculated metrics only to explain them; it does not invent live prices. Optional screening labels must state their rules and abstain when data are stale or insufficient. Historical evaluation should use time-aware splits, baseline comparisons, and measures such as precision, recall, F1, response time, and usability.

IV. CONCLUSION

The proposed chatbot combines stock search, historical charts, watchlists, and conversational explanations in one application. Its value is an accessible, data-grounded research workflow, not a promise of profitable prediction. Future work may add broader market coverage, licensed news sentiment, and deeper backtesting while keeping source information, uncertainty, and risk communication explicit.

FUNDING DETAILS

No external funding has been declared for this student project.

DISCLOSURE STATEMENT

The authors declare no known financial or personal conflict of interest related to this project. The application is an academic decision-support project and does not provide a guarantee of investment performance.

REFERENCES

[1] H. Markowitz, “Portfolio Selection,” The Journal of Finance, vol. 7, no. 1, pp. 77–91, Mar. 1952, doi: 10.1111/j.1540-6261.1952.tb01525.x.

[2] E. F. Fama, “Efficient Capital Markets: A Review of Theory and Empirical Work,” The Journal of Finance, vol. 25, no. 2, pp. 383–417, May 1970, doi: 10.1111/j.1540-6261.1970.tb00518.x.

[3] M. M. Kumbure, C. Lohrmann, P. Luukka, and J. Porras, “Machine learning techniques and data for stock market forecasting: A literature review,” Expert Systems with Applications, vol. 197, art. 116659, 2022, doi: 10.1016/j.eswa.2022.116659.

[4] K. Olorunnimbe and H. Viktor, “Deep learning in the stock market—a systematic survey of practice, backtesting, and applications,” Artificial Intelligence Review, vol. 56, pp. 2057–2109, 2023.