Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

FinTradeSim: Java-Based FinTech Platform for Paper Trading and Predictive Market Analytics with AI Agent

Prof. Ashwini Dake
JSPM's Jayawantrao Sawant College of Engineering, Pune

Kshitij Jagtap
JSPM's Jayawantrao Sawant College of Engineering, Pune

Ketan Chavan
JSPM's Jayawantrao Sawant College of Engineering, Pune

Krunali Pujari
JSPM's Jayawantrao Sawant College of Engineering, Pune

Rahila Wankhade
JSPM's Jayawantrao Sawant College of Engineering, Pune

Abstract

Financial markets have become increasingly accessible through digital trading platforms, yet beginners and students often face significant challenges in understanding market behavior, analyzing financial data, and making informed trading decisions without risking real capital. Existing paper trading platforms provide simulated trading environments but frequently lack integrated predictive analytics, intelligent decision support, and contextual financial guidance. This paper presents **FinTradeSim**, a Java-based FinTech platform designed to provide a comprehensive and risk-free learning environment for stock market participation. The proposed system integrates real-time market data, virtual portfolio management, predictive market analytics, sentiment-aware financial insights, and an AI-powered financial assistant within a unified architecture.

The platform utilizes live market information obtained through external financial APIs to simulate realistic trading conditions and supports technical analysis for market trend evaluation. An AI-driven query system based on Retrieval-Augmented Generation (RAG) combines financial news retrieval, vector-based knowledge storage, and large language model inference to deliver context-aware responses to user queries. The system is developed using Java-based web technologies, MySQL for data management, and modular API-driven integration to ensure scalability and maintainability.

By combining paper trading, market analytics, and AI-assisted financial support in a single platform, FinTradeSim bridges the gap between theoretical financial learning and practical market experience. The proposed system enables users to develop trading skills, understand market dynamics, and evaluate trading strategies without financial risk, thereby providing an accessible and educational environment for aspiring traders and investors.

Keywords

FinTradeSim, Paper Trading, FinTech, Predictive Market Analytics, AI Agent, Retrieval-Augmented Generation, Stock Market Simulation, Sentiment Analysis, Financial Education.

I. NTRODUCTION

1.1 Background

The rapid growth of digital trading platforms has significantly transformed participation in financial markets, making stock trading more accessible to a broader population of users through web-based and mobile technologies. At the same time, this transformation has been supported by scalable software ecosystems and API-driven integration models that enable modern financial applications to connect market data, analytics engines, and user-facing services in a unified environment [11][10][9].

Alongside this technological shift, the rise of retail investors has increased demand for systems that do more than simply execute trades. New participants often require tools that help them understand market behavior, interpret financial signals, and practice decision-making before risking real capital. This need has made financial literacy and simulation-based learning increasingly important within contemporary FinTech platforms [5][11].

Recent research in artificial intelligence has further expanded the possibilities for financial learning platforms. Machine learning models have shown strong potential for stock trend prediction using technical indicators and structured market data, while sentiment-aware models improve forecasting by incorporating financial news and textual context. In parallel, large language model based financial agents and multi-agent systems have demonstrated promising capabilities in financial reasoning, simulation, and decision support, opening the door to more interactive and educational trading environments [12][13][8][4][5].

1.2 Problem Statement

Although financial markets are more accessible than ever, real trading remains inherently risky, especially for beginners who may lack practical knowledge of market behavior, technical indicators, portfolio management, and disciplined decision-making. Entering live markets without prior experience can lead to financial losses and poor investment habits, making experimentation in real-money environments unsuitable for early-stage learners [7][12].

Another major challenge is the difficulty beginners face in interpreting fragmented market information. Predictive analytics, chart indicators, financial news, and advisory content are often distributed across separate applications or services, forcing users to switch repeatedly between platforms to make even basic decisions. This fragmentation weakens the learning process and reduces the effectiveness of market simulation as an educational activity [9][5].

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

Existing systems also tend to separate paper trading from intelligent assistance. Many paper trading platforms offer order execution simulation but provide limited support for contextual explanation, predictive insight generation, or AI-guided learning. As a result, users may be able to place simulated trades, but they are not adequately supported in understanding why a trade may be reasonable, risky, or mistimed [8][4].

1.3 Motivation

The motivation behind FinTradeSim arises from the need for an integrated learning-oriented platform that combines simulation, analytics, and intelligent guidance in a single environment. A unified platform can help users move beyond isolated actions such as buying or selling a stock and instead support a fuller learning cycle that includes observation, analysis, execution, and reflection [5][9].

AI-assisted analysis is particularly important because modern financial decision-making increasingly depends on the ability to synthesize structured and unstructured information. Research on sentiment-augmented forecasting, mixture-of-expert financial language models, and agentic AI in finance shows that combining multiple information sources can improve both prediction quality and user support capabilities. This creates a strong foundation for integrating predictive analytics and conversational financial assistance into an educational trading platform [13][8][4].

There is also a clear need for realistic real-time simulation. Reinforcement learning and simulation-oriented financial AI studies emphasize that dynamic market behavior, transaction logic, and evolving context are essential for meaningful experimentation and decision support. A platform that offers market-connected paper trading with analytics and AI explanations can therefore serve both educational and system design objectives [3][7][5].

1.4 Proposed Solution

To address these challenges, this paper proposes **FinTradeSim**, a Java-based FinTech platform designed for paper trading, predictive market analytics, and AI-assisted financial learning. The system is intended to provide users with a safe and structured environment in which they can simulate trades, monitor portfolio performance, analyze stock behavior, and interact with an AI financial assistant without exposing themselves to real-market losses [10][5].

FinTradeSim brings together multiple major modules within a unified architecture. These include user management, paper trading execution, portfolio monitoring, stock analytics, news-based contextual processing, and an AI assistant capable of answering financial queries using retrieved context and language model reasoning. Such an integrated design addresses the common separation between trading tools, analytical systems, and educational guidance found in many existing solutions [4][8][3].

From a technical perspective, the proposed system is grounded in scalable software and service-integration principles relevant to financial platforms. Java remains widely valued in financial software for reliability, security, and maintainability, while API integration enables communication with live data providers and AI services. This combination makes FinTradeSim suitable as both an academic system prototype and a practical foundation for future FinTech development [10][9].

1.5 Contributions

This work makes several contributions to the design of intelligent educational trading platforms. First, it proposes a real-time paper trading environment that enables users to simulate stock market activity, manage portfolios, and observe the outcomes of trading decisions without financial exposure. This supports experiential learning while avoiding the risks associated with real-money trading [1][5].

Second, the system incorporates predictive analytics informed by machine learning and market indicators. Prior studies show that Random Forest based approaches, sentiment-enhanced prediction models, and broader comparative machine learning analyses can provide useful foundations for financial forecasting modules. FinTradeSim adapts these ideas toward an applied learning platform rather than a standalone prediction engine [6][12][13].

Third, the platform integrates an AI financial assistant to improve user interaction and market understanding. Research on financial LLM systems and agentic AI suggests that intelligent assistants can support reasoning, contextual interpretation, and guided decision-making in finance-oriented environments. By embedding these capabilities into a unified architecture, FinTradeSim contributes a framework that connects paper trading, analytics, and AI-assisted learning in a single platform [8][4][5].

II. LITERATURE REVIEW

2.1 Machine Learning for Stock Market Prediction

Machine learning has become one of the most widely studied approaches for stock market prediction because it can model complex and non-linear relationships in financial data more effectively than traditional statistical techniques. In particular, Random Forest-based methods have demonstrated strong predictive capability by combining multiple decision trees, reducing overfitting, and leveraging technical indicators such as RSI, MACD, stochastic oscillator, and on-balance volume for direction prediction. Comparative studies in algorithmic trading also show that machine learning models can achieve meaningful forecasting performance when trained on carefully engineered features and evaluated using financial metrics such as accuracy, return, and risk-adjusted measures.[1][6]

An important improvement over standard ML approaches is sentiment integration. The SARF model enhances Random Forest prediction by incorporating financial news sentiment alongside technical indicators, showing improved robustness in volatile markets. This demonstrates that combining structured market data with unstructured textual information can significantly improve prediction quality in real-world conditions.[2]

Despite strong performance improvements, machine learning models still struggle with instability across market regimes. Financial data is noisy, non-stationary, and highly sensitive to external shocks, making long-term generalization difficult. Additionally, most research focuses on predictive accuracy rather than usability, interpretability, or integration into learning-based trading systems, leaving a gap between research models and practical educational platforms.[1][6][2]

2.2 Reinforcement Learning in Trading Systems

Reinforcement learning (RL) has emerged as a powerful paradigm for automated trading because it enables sequential decision-making through interaction with financial environments. Unlike supervised learning models that predict prices, RL agents learn optimal policies for actions such as buy, sell, or hold by maximizing cumulative reward over time.[4][5]

Recent research highlights that RL-based trading systems can adapt to changing market conditions by learning state representations and reward-driven strategies. However, realistic financial constraints such as transaction costs significantly impact performance, showing that theoretical profitability often decreases when applied in real-world conditions.[5][4]

Despite its promise, RL in trading remains challenging due to unstable training dynamics, reward sensitivity, and poor generalization across different market regimes. These limitations suggest that RL is best suited for simulation and decision-support

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

environments rather than fully autonomous trading systems, especially in educational platforms where interpretability is important.[4][5]

2.3 Large Language Models and Agentic AI in Finance

Large language models (LLMs) have introduced a new paradigm in financial systems by enabling reasoning over structured and unstructured data sources, including news, financial reports, and market indicators. TradExpert introduces a mixture-of-experts LLM architecture where specialized models handle different financial reasoning tasks and a final expert aggregates outputs for decision-making.[3][7]

StockAgent further extends this approach by simulating financial markets using multiple interacting agents powered by LLMs. This allows more realistic modeling of investor behavior, external events, and market dynamics. Related research in agentic AI systems shows that coordinated AI agents can support financial analysis, monitoring, and decision support while also introducing risks related to governance and model reliability.[7][8]

However, LLM-based trading systems face limitations such as latency, inconsistency, and sensitivity to prompting strategies. Despite these challenges, they are highly valuable for educational platforms because they can provide explanations, reasoning support, and contextual financial insights that traditional systems lack.[3][7][8]

2.4 Financial Technology Infrastructure

Modern financial systems depend heavily on robust software infrastructure. Java remains a widely used technology in fintech applications due to its scalability, security, and enterprise-level stability. These characteristics make it suitable for systems requiring reliable transaction handling and modular architecture.[10]

API integration plays a critical role in connecting different system components such as market data providers, analytics modules, and AI services. Well-designed APIs enable modular development and allow financial platforms to integrate multiple services efficiently without tight coupling.[11]

Blockchain-based financial systems further enhance transparency and automation through smart contracts and decentralized execution. However, scalability and integration challenges limit their use as a core trading infrastructure. Instead, blockchain concepts are often used as architectural inspiration for improving transparency and trust in financial systems.[9]

2.5 AI Inference Technologies

AI inference infrastructure is crucial for real-time financial applications where response latency directly affects user experience. Groq's Language Processing Unit (LPU) architecture is designed to optimize inference speed and energy efficiency, making it suitable for large language model deployment in time-sensitive environments.[12][13]

Low-latency inference is especially important in financial platforms that require real-time explanations, market insights, and interactive decision support. Faster inference improves system responsiveness and enhances the usability of AI-assisted trading tools.[13][7]

However, inference hardware alone does not guarantee system performance. Output quality still depends on model design, data pipelines, and integration architecture. Therefore, inference technologies should be viewed as enabling components within a broader financial AI ecosystem rather than standalone solutions.[12][13]

2.6 Comparative Analysis of Existing Systems

Existing research systems typically focus on isolated capabilities such as prediction, simulation, or reasoning rather than integrating all functionalities into a unified platform. The comparison below summarizes this limitation across representative systems.[1][7]

System	**Paper Trading**	**Analytics**	**AI Assistant**	**Real-Time Data**
Random Forest model	No	Yes (technical indicators)	No	Historical only [1]
SARF	No	Yes (sentiment + technical)	No	Market + news data [2]
RL trading framework	Simulation only	Yes (policy optimization)	No	Environment-based [4]
DRL forex model	Simulation/ backtesting	Yes	No	Sequence-based data [5]
TradExpert	No	Yes	Yes	Multi-source financial data [3]
StockAgent	Yes (simulated)	Yes	Yes	Event-driven simulation [7]

2.7 Research Gap

A clear gap exists in current literature. Machine learning research focuses on prediction accuracy, reinforcement learning emphasizes optimization strategies, and LLM-based systems focus on reasoning or simulation, but these are rarely combined into a single integrated educational platform.[4][3][1]

Additionally, most systems lack an educational orientation. They either generate predictions or simulate trading environments without explaining decision logic in a beginner-friendly way. This creates a gap between advanced financial AI systems and learner-centric platforms.[7][5]

FinTradeSim addresses this gap by integrating paper trading, predictive analytics, news-based context processing, and AI-driven explanation into a unified system. This enables both practical simulation and conceptual understanding within a single platform architecture.[3][8][7]

III. PROPOSED SYSTEM

3.1 System Overview

FinTradeSim is a Java-based FinTech platform designed to integrate paper trading, predictive analytics, and AI-assisted financial learning within a unified ecosystem. The system enables users to simulate stock trading in a risk-free environment while simultaneously receiving analytical insights, portfolio feedback, financial news context, and AI-driven explanations to support decision-making and learning.

The design is based on the principle that financial understanding improves when execution, analysis, and explanation are combined

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

in a single workflow rather than separated into independent tools. Therefore, FinTradeSim functions not only as a trading simulator but also as an educational platform that helps users interpret market behavior through structured analytics and intelligent assistance.

From an implementation perspective, the system follows a modular Java-based architecture. Backend services ensure reliability, concurrency control, and secure data handling, while API integration enables communication with external market feeds, news services, and AI inference systems. This supports extensibility and scalability for future enhancements.

3.2 System Architecture

The architecture of FinTradeSim consists of five layers: User Interface, Backend Services, Database Layer, AI Service Layer, and External API Layer. This layered design follows established fintech system principles where responsibilities are separated to improve maintainability and scalability.

The User Interface handles dashboards, trading actions, portfolio visualization, and AI interaction. Backend Services process requests, execute business logic, validate trades, and coordinate system modules. This separation ensures modular development and reduces system coupling.

The Database Layer stores user data, transactions, holdings, and market-related information. The AI Service Layer handles retrieval, context processing, and inference-based responses, while the External API Layer connects the system to live market feeds, news data, and inference engines.

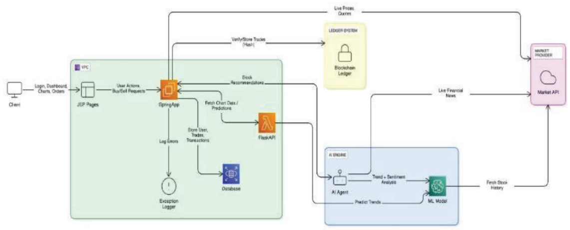

Figure 3.1 System Architecture Diagram

System flow includes:
• UI: JavaScript-based dashboards for trading and analytics
• Backend: Java/Spring services handling logic and API orchestration
• Database: MySQL for persistent storage
• AI Service: Python-based retrieval and LLM processing
• External APIs: Market data + news APIs + inference services

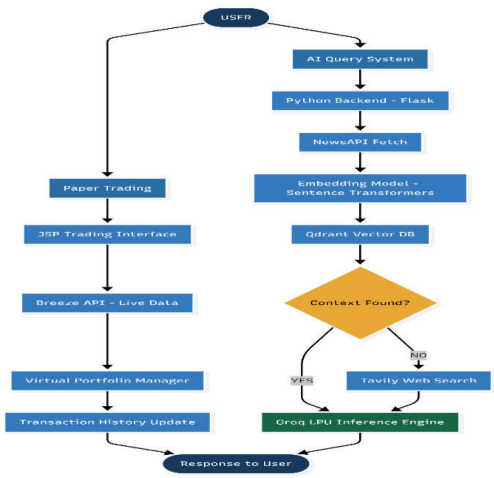

Figure 3.2 Flowchart

3.3 Functional Modules

3.3.1 User Management Module

This module handles user registration, authentication, and session management. It ensures secure access to personalized dashboards, trading simulations, and portfolio data. Secure identity handling is a fundamental requirement in fintech systems to ensure data integrity and controlled access.

It also maintains user profiles and session continuity, linking trades, watchlists, and AI interactions to individual accounts.

3.3.2 Paper Trading Module

The Paper Trading Module simulates buy/sell operations using real or near-real market data without real capital exposure. It processes orders, updates virtual portfolios, and records transaction history for analysis.

This module reflects simulation principles in reinforcement-based trading environments, where market interaction is modeled to replicate real trading conditions as closely as possible.

3.3.3 Portfolio Management Module

This module tracks holdings, profit/loss, and trade performance over time. It provides users with a consolidated view of investments and supports performance analysis for learning purposes.

It also enables users to connect trading actions with outcomes, improving understanding of market behavior in a structured way.

3.3.4 Stock Analytics Module

This module performs technical analysis using indicators and trend evaluation methods. Structured feature-based analysis can help identify meaningful market patterns when properly engineered.

Within FinTradeSim, the module is designed to support learning by allowing users to interpret signals rather than relying solely on automated predictions.

3.3.5 AI Financial Assistant Module

This module provides natural language interaction for financial queries. It uses retrieval-based and LLM-based reasoning to generate contextual responses about stocks, trading strategies, and market behavior.

It acts as a guided assistant rather than a general chatbot, helping users understand financial concepts and interpret trading outcomes within the platform context.

3.3.6 News Processing Module

This module collects financial news, analyzes sentiment, and integrates it into stock interpretation workflows. Combining sentiment signals with technical indicators improves market understanding and interpretation quality.

In FinTradeSim, this module enhances both analytics and AI responses by providing contextual awareness of external market events.

IV. METHODOLOGY

4.1 Development Methodology

The development methodology of FinTradeSim follows a modular workflow in which data acquisition, processing, simulation, analytics, and AI-assisted interaction are organized as connected but separable stages. This approach is consistent with modern API-integrated systems, where modular services allow applications to scale, evolve, and coordinate multiple specialized functions without excessive coupling.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

The methodology also reflects lessons from financial AI literature, where simulation quality depends on structured workflows, realistic information flow, and clearly defined stages for decision support and response generation. Rather than designing the platform as a single monolithic application, FinTradeSim is developed as a coordinated system of interoperable modules serving educational and analytical purposes together.

4.2 Data Acquisition

The system acquires market data through external financial APIs and news sources, enabling support for real-time or near-real-time simulation and contextual analysis. API integration is critical in such environments because it allows distributed systems to exchange current data, connect modular services, and maintain a flexible architecture.

The use of external APIs ensures that trading simulation and stock analysis are grounded in live information rather than only historical static datasets. This improves educational realism, as users can observe how changing market and news conditions influence simulated trades and analytical outcomes.

4.3 Data Processing

Once data is acquired, it passes through validation, cleaning, and storage stages before being used by application modules. This improves consistency by removing incomplete or malformed records, normalizing values, and ensuring downstream services such as analytics and AI retrieval operate on structured and reliable inputs.

Data processing is especially important in financial systems because prediction, simulation, and user-facing explanations depend heavily on data quality. Feature engineering and preprocessing significantly influence model performance and interpretability.

4.4 Paper Trading Workflow

The paper trading workflow begins with a user action such as placing a buy or sell request. The system validates the order, checks constraints, executes the simulated transaction, updates portfolio records, and logs transaction history. This structured flow reflects realistic trading environments where execution and state updates occur sequentially.

A well-defined workflow is essential in educational systems because it preserves the causal relationship between user decisions and outcomes. This allows users to analyze their trading behavior more effectively within the simulation environment.

4.5 Predictive Analytics Workflow

The predictive analytics workflow involves collecting market attributes such as price, volume, and technical indicator inputs. The system then computes indicators, performs trend analysis, and generates signals for user interpretation.

Structured feature engineering and hybrid signal integration improve market interpretation and predictive capability. In FinTradeSim, the objective is not only automation but also making analytical outputs interpretable for learning purposes.

4.6 AI Query Processing Workflow

The AI query processing workflow begins when a user submits a financial question through the assistant interface. The system converts the query into embeddings, retrieves relevant contextual information, and generates responses using large language model inference. This retrieval-augmented generation approach is widely used in modern financial AI systems for improving contextual accuracy.

Such workflows are effective because they combine structured retrieval with generative reasoning, allowing more grounded and relevant responses compared to standalone language models. Multi-source reasoning approaches further support financial decision support and contextual understanding.

4.7 Database Management

The database design of FinTradeSim supports persistent storage of users, portfolios, transactions, watchlists, and news data. These structured entities form the backbone of system state management and ensure consistency across modules.

Efficient database design is crucial for maintaining synchronization between trading simulation, analytics, and AI-driven responses. A well-structured database enables smooth integration of user behavior, market data, and contextual insights across the system.

V. SYSTEM IMPLEMENTATION

5.1 Technology Stack

The implementation of FinTradeSim is based on a layered technology stack designed to support modular development, secure transaction processing, API interoperability, and AI-assisted financial analysis. Java is used as the primary backend technology due to its strong adoption in financial systems, offering reliability, scalability, concurrency support, and secure execution for transaction-heavy applications. API-driven integration is also essential because the system depends on coordinated communication between market data providers, news services, storage systems, and AI-based inference services.

The architecture reflects a hybrid implementation model in which Java handles core business logic and transactional workflows, while Python-based services manage AI processing, retrieval pipelines, and model inference tasks. This separation follows modern distributed system design principles, where specialized services communicate through APIs instead of being tightly coupled within a single runtime environment.

Layer	Technology
Frontend	JSP, JavaScript, CSS
Backend	Spring Boot
Database	MySQL
AI Service	Flask
Vector Database	Qdrant
LLM	Groq
APIs	Breeze API, NewsAPI

The frontend layer provides user interaction interfaces such as dashboards, stock views, portfolio tracking, and AI assistant access. The backend layer built using Spring Boot manages authentication, business logic, and API orchestration. MySQL is used for structured storage of users, trades, holdings, and system data. The AI service layer uses Flask for lightweight processing, Qdrant for vector-based retrieval, and Groq for fast inference-based response generation.

5.2 Database Design

The database design of FinTradeSim is structured to support persistent storage of users, portfolios, transactions, watchlists, and news data. In financial simulation systems, persistence is essential

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

for maintaining consistency across sessions, ensuring that trading history, holdings, and user interactions remain accurate and traceable.

The main entities include Users, Portfolio, Transactions, Watchlist, and News. The Users table stores authentication and profile data, while Portfolio maintains current holdings and balances. The Transactions table records all buy and sell operations performed during paper trading, enabling performance evaluation and historical analysis. Watchlist stores user-selected stocks for monitoring, and News stores processed financial information used by analytics and AI components.

A relational database structure is appropriate because these entities have clearly defined relationships and require consistency across updates. For example, each user may have multiple transactions, portfolios depend on executed trades, and AI modules may retrieve contextual data from both user activity and market news. This structured approach ensures data integrity and supports modular system integration.

5.3 API Integration

API integration is a core component of FinTradeSim because the system relies on external data sources rather than isolated internal datasets. APIs enable communication between distributed services and support modular, scalable system design in modern applications.

The Breeze API is used for retrieving market data such as stock prices and trading-related attributes. This data supports both paper trading execution and analytical modules, ensuring that simulation is based on current market conditions rather than static datasets. Real-time data integration improves the educational value of the system by allowing users to observe how market changes affect simulated decisions.

NewsAPI is used for financial news collection, enabling the system to incorporate external textual information into analytics and AI processing workflows. This supports sentiment-aware analysis and contextual interpretation of market behavior, which improves financial understanding when combined with structured market data.

Groq is integrated as the inference engine for the AI assistant module. It provides low-latency response generation for user queries after contextual retrieval is performed. Fast inference is critical in interactive financial systems where response delay can reduce usability and learning effectiveness.

5.4 Security Considerations

Security is a fundamental requirement in FinTradeSim due to its handling of user accounts, portfolio data, and external API interactions. Java-based systems are commonly used in financial applications because they provide stable runtime behavior and built-in security mechanisms suitable for enterprise-grade environments.

Authentication mechanisms ensure that only authorized users can access personal dashboards, trading modules, and portfolio data. Session management maintains continuity of user interactions while ensuring isolation between different accounts, preventing unauthorized access or data leakage.

API security is also critical because the system integrates external services such as Breeze API, NewsAPI, and Groq. API keys and credentials must be securely stored on the server side and never exposed in client-side code. Proper access control and backend-managed request handling reduce vulnerability exposure and ensure secure communication between system components.

VI. RESULTS

6.1 Dashboard Interface

The FinTradeSim dashboard serves as the central interface through which users can access market summaries, portfolio status, analytics outputs, and AI-assisted insights in one place. This unified presentation improves visibility, reduces fragmentation, and supports smoother interaction across multiple services and data sources.

The dashboard successfully integrates simulation, portfolio tracking, and analytical feedback into a single environment, allowing users to monitor their trading activity and interpret outcomes efficiently. It enables a structured view of financial activity, making it easier for users to understand market behavior and system responses in real time.

6.2 Stock Tracker

The stock tracker module enables users to monitor stock prices, observe short-term movements, and review market-related updates relevant to individual securities. It provides continuous visibility into price behavior and supports informed observation within the simulation environment.

The module allows users to track changes in stock performance over time and correlate them with trading decisions and market conditions. This improves awareness of market fluctuations and enhances the learning experience by providing consistent and structured stock-level information.

6.3 Paper Trading Module

The paper trading module provides a risk-free environment where users can place buy and sell orders, test strategies, and observe the outcomes of their trading decisions without real financial exposure. It simulates the complete trading workflow including order placement, execution, and portfolio updates.

The system ensures that each transaction is processed in a structured manner, allowing users to clearly observe how decisions translate into portfolio changes. This helps in understanding trading mechanics and improves practical learning through simulated market participation.

6.4 Portfolio Management

The portfolio management module tracks user holdings, evaluates profit and loss, and presents overall investment performance. It provides a consolidated view of assets and helps users monitor the outcomes of their trading activities over time.

The module enables users to understand how individual trades contribute to overall portfolio performance. It supports continuous evaluation by presenting updated financial status after each simulated transaction, improving decision awareness and learning outcomes.

6.5 AI Financial Assistant

The AI financial assistant module allows users to interact with the system using natural language queries related to stocks, trading strategies, and market behavior. It retrieves relevant context and generates explanatory responses to support user understanding.

The assistant enhances the platform by providing interpretative support for financial concepts and trading outcomes. It helps users understand market conditions, clarify doubts, and gain insights into simulated trading activities in an interactive manner.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

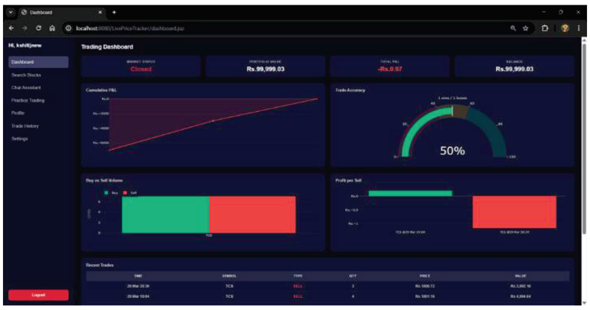

Figure 6.1 User Dashboard

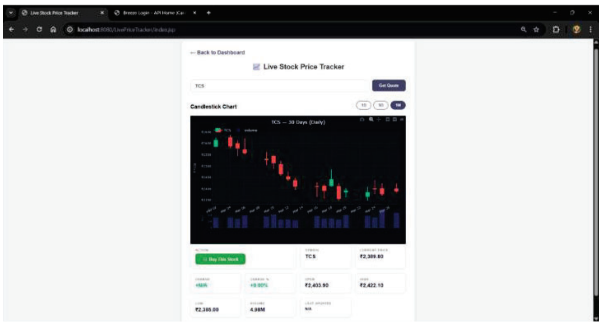

Figure 6.2 Stock Tracker

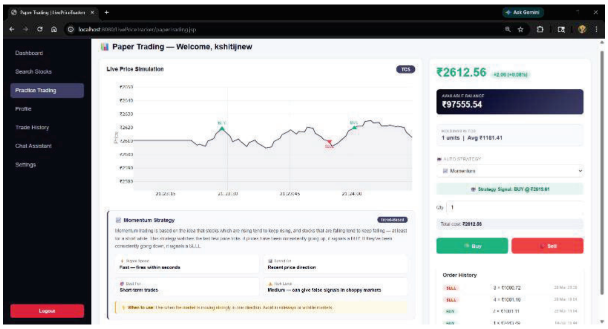

Figure 6.3 Paper Trading Module

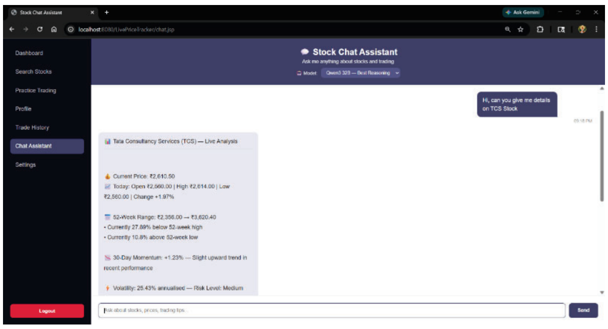

Figure 6.4.1 Chat Assistant

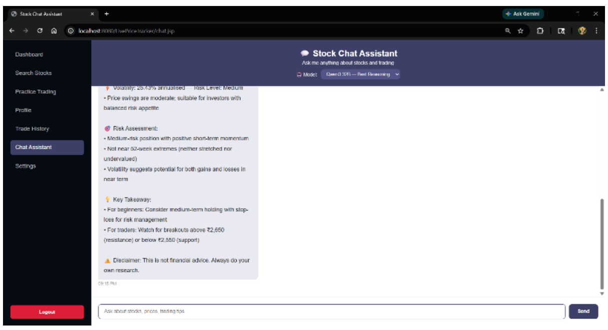

Figure 6.4.2 Chat Assistant

VII. CONCLUSION

This report presented FinTradeSim, a software-based FinTech platform designed to integrate paper trading, market analytics, and AI-assisted financial learning into a unified system. The primary objective of the system is to provide users with a structured environment where they can simulate stock trading without financial risk while simultaneously gaining analytical insights and contextual understanding of market behavior. By combining trading simulation, portfolio tracking, and intelligent assistance, the system addresses the gap between isolated trading tools and integrated learning-oriented financial platforms.

The proposed system effectively brings together multiple functional modules including paper trading, stock tracking, portfolio management, and an AI financial assistant into a single cohesive workflow. This integration allows users to not only execute simulated trades but also interpret outcomes through analytics and natural language explanations. The modular architecture ensures that each component operates independently while still contributing to a unified user experience.

The system demonstrates that combining simulation with analytical and AI-driven support improves the overall usability and educational value of financial platforms. Users are able to observe market behavior, understand the impact of their trading decisions, and receive contextual guidance in real time. This makes FinTradeSim suitable as both a learning tool for beginners and a practice environment for understanding real-world trading concepts.

Although the current implementation provides a functional and integrated system, it remains dependent on external APIs for market data and inference services. This introduces certain limitations related to availability and performance consistency. However, within these constraints, the platform successfully achieves its goal of delivering a cohesive and interactive financial simulation environment.

In future enhancements, the system can be extended by incorporating real-time trading simulation with higher data granularity, improved predictive analytics models, and more advanced AI reasoning capabilities. Additional improvements such as performance optimization, scalable cloud deployment, and enhanced personalization features can further strengthen the platform. Overall, FinTradeSim represents a meaningful step toward building intelligent, software-driven financial learning systems that combine simulation, analytics, and AI assistance in a single ecosystem.

VIII. REFERENCES

[1] L. Khaidem, S. Saha, and S. Roy Dey, “Predicting the direction of stock market prices using random forest,” Applied Mathematical Finance, 2016.
[2] S. Talazadeh and D. Peraković, “SARF: Enhancing stock market prediction with sentiment-augmented random forest,” 2024.
[3] Q. Ding, H. Shi, J. Guo, and B. Liu, “TradExpert: Revolutionizing trading with mixture of expert LLMs,” 2025.
[4] S. Yasin and P. S. Gill, “Reinforcement learning framework for quantitative trading,” 2024.
[5] K. Ishikawa and K. Nakata, “Online trading models with deep reinforcement learning in the Forex market considering transaction costs,” 2021.
[6] A. Jabbar and S. Q. Jalil, “A comprehensive analysis of machine learning models for algorithmic trading of Bitcoin,” 2024.
[7] C. Zhang et al., “When AI meets finance (StockAgent): Large language model-based stock trading in simulated real-world environments,” 2024.
[8] Okpala, A. Golgoon, and A. R. Kannan, “Agentic AI systems

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

applied to tasks in financial services modeling and model risk management crews," 2025.
[9] S. E. Chang and M.-H. Wang, "Blockchain-enabled FinTech innovation: A case of reengineering stock trading services," 2023.
[10] Reshko, "Java development in fintech digital payment systems," 2024.
[11] IBM, "What is API integration?," 2025.
[12] S. Raghavan, I. Arsovski, and D. Maheshwari, "The Groq LPU AI inference technology delivers more energy efficiency than GPUs for AI inference." 2024.
[13] Groq, "What is a Language Processing Unit (LPU)?," 2024.