JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Real-Time Market Transactions with Automated Pattern Recognition

[1]Rajnandini Kumawat, [2]Sumit Singh, [3]Gourav Chaturvedi, [4]Abdul Samad Kasu, [5]Mohit Tripathi

[1]*Professor*, [2]*Student*, [3]*Student*, [4]*Student*, [5]*Student*

Computer Department, Lokmanya Tilak College of Engineering

Navi Mumbai, India

Abstract : This project aims to develop a web application that simplifies stock market education for students through interactive learning modules, real-time stock monitoring, and a risk-free simulated trading environment. By integrating API-driven data retrieval, secure database management, and an intuitive user interface, the platform bridges the gap between theory and practice. Features like analytics, feedback, and community engagement enhance learning, making financial literacy more accessible and engaging.

IndexTerms **- Stock Market Education, Financial Literacy, Interactive Learning, Simulated Trading, Real-Time Stock Monitoring, API-Driven Data, Data Analytics, User Engagement, Web Application, Market Trends Analysis.**

I. INTRODUCTION

The stock market is a complex and dynamic financial environment that plays a critical role in the global economy. However, a significant portion of students and young learners lack foundational knowledge and practical understanding of how the stock market operates. Traditional financial education often fails to provide an engaging and interactive approach to learning about stock market concepts. Furthermore, tools that allow for monitoring live stock results are typically designed for advanced users, making them inaccessible to beginners.

II. Problem Identification

Students often struggle to grasp fundamental stock market concepts, such as stock trading mechanisms, price-influencing factors, and trading risks. This **knowledge gap** prevents them from making informed decisions, limiting their ability to learn through practical experience. The **lack of practical exposure** in stock market education arises because most resources are purely theoretical, offering little to no hands-on learning experiences or opportunities to track real-time market data. Additionally, **accessibility and engagement** in stock market education are hindered by current platforms that are not user-friendly for beginners, discouraging students from exploring this critical aspect of financial literacy. The **growing demand for financial literacy** has highlighted the need for tools that equip students with the skills required to understand and participate in the stock market, especially as financial independence and planning become more emphasized.

III. Literature Review

Comparison of Existing Educational Stock Market Platforms and Their Limitations

- Existing platforms like Robinhood and E*TRADE are designed for experienced traders, lacking beginner-friendly educational resources. Apps like Investopedia and Stock Market Simulator offer simulated trading but use delayed data and lack real-time insights, limiting hands-on learning.

Research on AI-Based Stock Market Education

- AI-powered platforms can provide personalized learning and decision-making insights, but are rarely implemented in educational stock market tools. This project integrates AI and machine learning models to analyze trends and guide users, offering a more interactive and insightful learning experience.

The Impact of Gamification on Financial Education and Learning Outcomes

- Gamification in financial education improves engagement and retention. By incorporating elements like rewards and challenges, it makes learning more interactive and enjoyable, which has shown positive impacts on student participation

and knowledge retention in financial literacy.

IV. Objective

To develop a web application that bridges the gap between theoretical learning and practical application in stock market education. The web app aims to:

1. Provide an interactive and beginner-friendly platform for learning stock market concepts.
2. Offer tools for monitoring live stock results and understanding real-time market dynamics.
3. Enhance user engagement through gamification, quizzes, and scenario-based learning.
4. Foster financial literacy by simplifying complex stock market processes for young learners.

V. Proposed Solution

This project aims to develop a web application that helps students learn about the stock market through a structured and engaging platform.

1. Interactive Learning Modules – The platform will simplify complex stock market concepts through interactive lessons, covering topics like types of stocks, market indices, trading strategies, risk management, and portfolio diversification.

2. Real-Time Stock Monitoring – Live stock market data will be integrated using APIs, enabling users to track stock performance, follow market trends, and analyze data over time.

3. Simulated Trading Environment – A virtual trading feature will allow students to practice investing without financial risk. Users will receive virtual funds to trade, experience market fluctuations, and develop decision-making skills.

4. Analytics and Feedback – The platform will provide insights into users' progress, offering reports on simulated trading performance and identifying areas for improvement.

5. Community and Collaboration – Discussion boards and forums will encourage students to ask questions, exchange ideas, and engage in collaborative learning

6. System Architecture

The architecture of the proposed web application consists of multiple interconnected components, ensuring seamless interaction between the frontend, backend, APIs, and databases.

High-Level Architecture Diagram

Frontend: Developed using React.js for an intuitive user experience, incorporating interactive stock charts, a virtual–≠ trading interface, and learning modules.

Backend: Built using Python (Django/Flask) or Node.js to handle user requests, stock data processing, and authentication.

APIs: yFinance and Alpha Vantage APIs are used to fetch real-time stock data, providing live updates and historical price trends.

Database: A PostgreSQL/MySQL database stores user profiles, transaction history, and learning progress. Firebase authentication is integrated for secure login.

User Authentication: Uses Firebase authentication or OAuth2.0 to enable secure sign-in using Google or email credentials.

API Requests: Frontend interacts with backend services via RESTful APIs, fetching live stock data, executing simulated trades, and providing analytics.

Database Queries: SQL-based queries handle user portfolio tracking, stock performance analysis, and trade history storage.

ML Model Training and Implementation

Machine learning models will enhance the system by analyzing market trends and making stock predictions.

1. Pattern Recognition Techniques

Moving Average: Identifies trends by averaging stock prices over time.

a. **SMA** (Simple Moving Average): It helps smooth price fluctuations to identify trends.

$$SMA = (\sum_{i=1}^{N} \times P_i) \div N$$

b. **EMA** (Exponential MOving Average): It gives more weight to recent prices for quicker trend detection.

$$EMA_t = (P_t \times (\frac{2}{N+1})) + (EMA_{t-1} \times (1 - \frac{2}{N+1}))$$

where,

P_i = *Price of the stock at time i*
N = Number of periods
EMA_{t-1} *= Previous EMA Value*

Relative Strength Index (RSI): Detects overbought or oversold stocks based on price momentum. **Determines whether a stock is overbought (>70) or oversold (<30).**

$$RSI = 100 - (\frac{100}{1+RS})$$

where,

RS = average gain / average loss over a specified period (usually 14 periods)

Moving Average Convergence Divergence (MACD): Measures trend direction and strength by comparing moving averages. **Helps identify momentum changes.**

$$MACD = EMA_{12} - EMA_{26}$$

where,

EMA_{12} = *12-period EMA*
EMA_{26} *= 26-period EMA*

2. Predictive Analysis Models

I. LSTM (Long Short-Term Memory): A deep learning model for time-series forecasting of stock prices. LSTM consists of three gates:

a. **Forget Gate:**

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

where,

$f_t \rightarrow$ *Forget gate activation*
$W_f , b_f \rightarrow$ *Weight matrix and bias for the forget gate*
$h_{t-1} \rightarrow$ *Previous hidden state*
$x_t \rightarrow$ *Current input*
$\sigma \rightarrow$ *Sigmoid activation function*

b. **Input Gate:**

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\check{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \check{C}_t$$

where,

$i_t \rightarrow$ *Input gate activation*
$\check{C}_t \rightarrow$ *Candidate cell state*
$C_t \rightarrow$ *Updated cell state*
$W_i , W_C , b_i , b_C \rightarrow$ *Weight matrices and biases.*

c. **Output Gate:**

$$o_t = \sigma(W_o . [h_{t-1}, x_t] + b_o)$$

where,

$o_t \rightarrow$ *Output gate activation*

$h_t \rightarrow$ *New hidden state*

Data Flow and Preprocessing

Real-time stock data needs efficient handling and processing for seamless user experience

1. **Fetching, Storing, and Displaying Data**

   Fetching: The system calls yFinance API every 5 seconds to update stock prices.

   Storing: Only relevant stock data (symbol, price, volume) is stored in a PostgreSQL database to minimize storage overhead.

   Displaying: Frontend dynamically updates stock charts using WebSockets for real-time updates.

2. **Data Handling Mechanisms**

   Caching: Redis is used to cache frequently requested stock prices, reducing API request loads

$$Change\ \% = \left(\frac{Closing\ Price - Opening\ Price}{Opening\ Price}\right) \times 100$$

Volatility Calculation is performed for risk analysis is performed using:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (r_i - \underline{r})^2}$$

where,

r_i = Individual return

$\underline{r}$ = Mean return

N = Number of observations

σ = Standard Deviation (Volatility)

Reducing Latency: API requests are optimized using batch processing, reducing response time to under 100ms.

Preprocessing: Data is cleaned to remove null values, outliers, and inconsistencies before being used for ML predictions. Stock price changes are calculated to measure trends.

API Selection

Q. Why yFinance?

A. To retrieve real-time stock market data, **yFinance** was selected due to its numerous advantages:

1. **Free and Open-Source**: yFinance provides stock data at no cost, making it an ideal choice for a student-focused educational platform.
2. **Comprehensive Data**: Offers historical stock prices, financial statements, and market indicators essential for predictive modeling.
3. **Easy Integration**: Compatible with Python and commonly used machine learning libraries.
4. **Reliable and Updated Data**: Fetches real-time and historical stock data directly from Yahoo Finance, ensuring accuracy.
5. **Scalability**: Suitable for expanding the platform to handle more stock data as the user base grows.

- **Commission) Guidelines**: Adheres to ethical stock data presentation and prevents misleading information.

7. Testing and Evaluation

Before deployment, rigorous testing ensures smooth operation.

Unit Testing

API responses are tested for **accuracy** (stock prices match real-time values).

- Authentication is tested with **multiple login scenarios** to ensure security.
- UI elements are tested for **responsiveness across devices**.
-

Performance Testing

- API response time is optimized to be **under 200ms**.
- Load testing simulates **10,000 concurrent users** to measure scalability.
- Virtual stock trades execute with **<1s delay** for a seamless experience.

Expected Impact

- **Enhanced Financial Literacy**: Students will gain a better understanding of stock market principles, preparing them for informed financial decision-making.
- **Practical Skills Development**: Users will acquire hands-on experience in analyzing market trends and making trading decisions, enabling them to apply their knowledge in real-world scenarios.
- **Increased Accessibility**: The web app will make financial education more approachable and engaging, encouraging more students to explore the stock market.
- **Long-Term Benefits**: By fostering an early interest in financial markets, the application can contribute to creating a generation of financially aware and responsible individuals.

Challenges and Mitigation Strategies

- **Technical Complexity**: Developing features like real-time data monitoring and a simulated trading environment can be technically demanding. To address this, the project will utilize reliable APIs and modular development practices to streamline implementation.
- **User Engagement**: Ensuring that the app remains engaging and user-friendly for students of varying skill levels will be critical. Regular user feedback and iterative design updates will help maintain high engagement levels.
- **Scalability**: As the user base grows, ensuring that the platform can handle increased traffic and data load will be essential. Cloud-based solutions and scalable infrastructure will be employed to address this challenge.
- **Data Accuracy**: Reliance on third-party APIs for live stock data introduces the risk of inaccuracies or downtime. Using multiple data sources and implementing fallback mechanisms will mitigate these risks.

8. Conclusion

By addressing the challenges associated with stock market education, this web application has the potential to transform the way students learn about and engage with financial markets. The project aims to empower the next generation with the knowledge and tools needed to navigate the complexities of the stock market with confidence. Through interactive learning, real-time monitoring, and practical exposure, the web app will serve as a comprehensive platform for fostering financial literacy and encouraging informed participation in the stock market.

REFERENCES

1. **Robinhood**: Robinhood. (2024). “How to Trade Stocks: A Beginner’s Guide.” Retrieved from https://www.robinhood.com.

2. **E*TRADE**: E*TRADE. (2024). “Stock Market Education: Learning the Basics.” Retrieved from https://www.etrade.com.

3. **Investopedia**: Investopedia. (2024). “Stock Market Simulator: Simulating Real Market Conditions.” Retrieved from https://www.investopedia.com.

4. **Sahu, S., & Rath, M. (2019).** Financial Literacy Platforms for Young Investors: The Role of Online Education in Stock Market Learning. *Journal of Educational Technology and Financial Literacy, 27*(1), 98-112. Retrieved from https://www.jetfl.edu/.

5. **AI in Stock Market Education**: Zhang, H., & Lee, J. (2023). “AI-Based Stock Market Education: Applications and Challenges.” *Journal of Financial Technology*, 5(2), 23-42.

6. **Gamification in Financial Education**: Suresh, K. & Sharma, P. (2022). "Gamification for Financial Education: Enhancing Engagement and Learning Outcomes." *International Journal of Educational Technology*, 18(3), 65-79.

7. **Personalized Learning with AI**: Anderson, C., & Parker, A. (2022). "The Role of AI in Personalized Financial Education." *Journal of Financial Learning*, 7(1), 49-58.

8. **Avi, B., & Eisenberg, G. (2021).** Enhancing Financial Literacy through Digital Platforms: A Study on Stock Market Education. *Journal of Financial Education, 35*(2), 45-60. Retrieved from https://www.jfe.edu/.

9. **Liu, Y., & Zhang, J. (2020).** Gamification and Its Impact on Financial Learning Platforms. *International Journal of Financial Studies, 18*(3), 130-142. https://doi.org/10.1108/IJFS-12-2020-0345.