

Luminex: A Secure Full-Stack Paper Trading Platform for Financial Education

ADITYA SHASHIKANT INGOLE¹, MITESH KASHINATH WAKADE², BUDDHITA DAGADU BHAGAT³, ADITYA BHAGIRATH JAGTAP⁴

1,2,3,4 Department of Computer Engineering, Sinhgad Institute of Technology, Lonavala, Pune, India

Under the guidance of Dr. Shubhangi R. Patil

Abstract- *Financial literacy among students and beginner investors remains critically low, and the tools available to address this gap are either too theoretical or too insecure to serve as practical reference implementations. Luminex was developed to bridge this gap. It is a full-stack, browser-based virtual stock trading platform built with Next.js 15, TypeScript, Node.js, Express.js, and MongoDB that simulates realistic Indian equity and derivatives market conditions for educational purposes. The platform provides a structured, end-to-end trading workflow combining OTP-based email authentication, JWT session management, real-time simulated price updates on a 3-second refresh cycle, virtual buy and sell operations for equities and Futures & Options (F&O), watchlist management, candlestick charting across five timeframes, and dynamic portfolio analytics with sector-wise allocation - all protected by a multi-layer security architecture implementing bcrypt password hashing, rate limiting, and account lockout protection. The system follows a three-tier client-server architecture with eight well-defined MongoDB collections and a modular Express.js backend. Experimental evaluation across comprehensive unit, integration, system, and security test suites demonstrates full correctness of all modules with API response times under 150 ms and OTP delivery under five seconds. Beyond its practical capabilities as a trading simulator, Luminex serves as a production-grade full-stack reference implementation that demonstrates modern secure web application design principles directly applicable to industry software development roles.*

Index Terms- *Paper Trading, Virtual Stock Market Simulation, Next.js, Node.js, Express.js, MongoDB, JWT Authentication, OTP Verification, Portfolio Management, Futures and Options, Fintech Education, Secure Web Application, bcrypt, Real-Time Simulation, Full-Stack Development*

I. INTRODUCTION

Financial illiteracy remains a persistent barrier to wealth creation for millions of young Indians. College curricula in finance and computer engineering rarely provide interactive, hands-on tools for developing practical market intuition. Students who understand the theoretical mechanics of equity trading, options pricing, or portfolio rebalancing frequently lack the practical experience required to act confidently in live markets. The obvious solution - practising with real money - carries the financial risk that makes such practice untenable for most learners.

Paper trading simulators exist to address this gap, but they rarely do so adequately. A survey of freely available platforms reveals a consistent pattern: either the platform provides realistic market features without adequate security implementation, or it implements basic security without the full feature set a serious learner requires. Investopedia Simulator offers virtual trading but lacks derivatives support and modern authentication. NSE Paathshaala covers equity simulation but presents an outdated interface without JWT-based session security. TradingView's paper trading feature provides excellent charting but omits portfolio analytics. None provides an integrated, production-grade reference architecture that is itself instructive for students of computer engineering.

Luminex was designed to address all of these gaps simultaneously. Rather than treating security and trading functionality as independent concerns, the platform integrates them into a single cohesive system built on modern full-stack web development technologies that directly mirror current industry practice. The result is a platform that serves two distinct but complementary educational purposes: it teaches financial market mechanics through realistic simulation, and it teaches secure full-stack development through its own architecture.

Three principles guided the design. First, realism: simulated market behaviour should closely reflect actual Indian equity and F&O market mechanics, including candlestick charting, options chains with Greeks, sector-wise portfolio allocation, and real-time price simulation. Second, security: authentication, session management, and API protection should meet production-grade standards, implementing OWASP-recommended mechanisms. Third, completeness: the platform should cover the full trading lifecycle from registration through watchlist curation, order placement, position management, and portfolio analytics, without requiring any external paid data service.

This paper presents the design, architecture, implementation, and evaluation of the Luminex platform.

II. LITERATURE REVIEW

A. Existing Paper Trading Platforms

Commercial and educational trading simulators such as Investopedia Simulator, Moneybhai, and NSE Paathshaala have made virtual trading accessible to a broader audience [1]. However, as shown in Table I, they share structural limitations: absent or minimal security features, no derivatives trading capability, outdated user interfaces, or a gamified approach that sacrifices realism. Moneybhai, for example, operates as a community game rather than a realistic simulation, trading accuracy for engagement metrics.

The absence of modern authentication mechanisms - OTP-based email verification, JWT session tokens, and bcrypt password hashing - is particularly concerning for platforms that store user data. A simulation platform that itself demonstrates insecure practices is unsuitable as a reference implementation for computer engineering students.

Table I: Comparison of Existing Paper Trading Platforms

| Platform | F&O | JW T Auth | Real-time | Analytic s |
|--------------------|---------|-----------|-----------|------------|
| Investopedia Sim. | X | X | Partial | Basic |
| NSE Paathshaala | X | X | X | X |
| TradingView Paper | Partial | N/A | Yes | X |
| Moneybhai | X | X | Delayed | X |
| Luminex (Proposed) | Yes | Yes | Yes | Yes |

B. Secure Authentication in Web Applications

Sharma's study on secure authentication in financial systems identified multi-factor authentication - and specifically OTP-based verification - as essential for any platform handling sensitive user information [2]. The OWASP Foundation's 2023 guidelines recommend bcrypt with a minimum cost factor of ten for password hashing, rate limiting to prevent brute-force attacks, and account lockout after repeated failed attempts [3]. Luminex implements all three mechanisms as first-class architectural concerns.

C. JWT as a Stateless Authentication Standard

RFC 7519 defines the JSON Web Token standard as a compact, self-contained mechanism for stateless authentication [4]. JWT's stateless nature eliminates server-side session storage overhead and supports horizontal scalability. Luminex implements a dual-token architecture: a short-lived access token (15-minute validity) and a long-lived refresh token (7-day validity), signed using HMAC-SHA256, consistent with best practices for web financial applications.

D. Educational Trading Simulators and Financial Literacy

Brown's study of educational trading simulators demonstrated that realistic market simulation significantly improves financial literacy outcomes compared to theoretical-only instruction [5]. Students exposed to hands-on virtual trading showed higher retention of concepts including portfolio diversification, options risk mechanics, and the relationship between volatility and price. This finding motivates Luminex's investment in realistic simulation fidelity - including options Greeks, synthetic implied volatility, and sector-wise allocation charts.

E. Research Gap

The literature consistently identifies a gap: no freely available platform integrates OTP authentication, JWT session management, real-time F&O simulation, candlestick charting, and portfolio analytics in a single secure, modular, and extensible architecture. Luminex closes this gap.

III. METHODOLOGY

1) Overall System Design

Luminex follows a Three-Tier Client-Server Architecture. The Presentation Layer is built with Next.js 15 and TypeScript. The Application Layer uses Node.js with Express.js, housing all business logic, authentication workflows, and trading engine. The Data Layer uses MongoDB with Mongoose ODM, storing eight distinct collections. Communication between tiers

occurs exclusively via RESTful HTTP APIs, ensuring clean separation of concerns and enabling each layer to be tested independently.

2) Authentication Module

The authentication module implements a three-stage verification process. During registration, user data is validated server-side, the password is hashed using bcrypt with a cost factor of ten, and the user record is persisted with isVerified=false. A six-digit OTP is then generated, stored in MongoDB with a 10-minute TTL, and delivered via Nodemailer SMTP. Upon correct OTP entry within the window, the user is marked verified and JWT access (15-minute validity) plus refresh tokens (7-day validity) are issued and stored as HTTP-only cookies. JWT token integrity is verified using HMAC-SHA256.

3) Market Simulation Engine

The market simulation engine generates realistic intraday price fluctuations using a seed-based random walk algorithm. Every three seconds, a new price is computed:

$$P_{\text{new}} = P_{\text{base}} \times (1 + \text{direction} \times \text{volatility})$$

where volatility in [0, 0.02] (max +/-2%) and the price is clamped within +/-20% of its base value. The seeding approach uses a deterministic hash of the stock symbol with the current date string, ensuring reproducibility within a session. Account lockout is enforced after five failed attempts with a 30-minute cooldown. The API layer applies a sliding-window rate limiter (100 requests per 15-minute window).

4) Trading and F&O Module

The trading module supports virtual buy and sell operations for ten NSE-listed equities. On order placement, the system validates the user's virtual cash balance against the trade value (price x quantity), executes the trade atomically, updates the holdings record, debits the user's virtual cash, and records the transaction in MongoDB. The F&O module presents an active futures tab and an options chain tab per symbol. Options Greeks - Delta and Theta - are computed synthetically. Synthetic implied volatility is shown as: $IV = \sqrt{(2 \cdot \pi / T)} \times C / S$, where C is the option premium, S is the spot price, and T is time to expiry in years.

5) Portfolio Management Module

Portfolio calculations execute server-side on every page load. The backend fetches all holdings, retrieves the current simulated LTP for each symbol, and computes:

$$P\&L = (LTP - \text{AvgBuyPrice}) \times \text{Quantity}$$

$$\text{Return (\%)} = \left[\frac{(\text{Current Value} - \text{Invested Value})}{\text{Invested Value}} \right] \times 100$$

Aggregated totals - total current value, total invested, and total P&L - are returned alongside a sector-wise allocation breakdown rendered as a pie chart on the frontend.

6) Technology Stack

Table II: Luminex Technology Stack

| Layer | Technology | Purpose |
|----------------|----------------------------------|---|
| Frontend | Next.js 15 + TypeScript | SSR React framework, type safety |
| Frontend | Tailwind CSS + shadcn/ui | Responsive utility styling, accessible components |
| Frontend | Lightweight Charts (TradingView) | Interactive candlestick charting |
| Backend | Node.js + Express.js | Server runtime, HTTP routing |
| Database | MongoDB + Mongoose ODM | Document store, schema enforcement |
| Auth | jsonwebtoken + bcryptjs | JWT issuance/verification, password hashing |
| Auth | Nodemailer (SMTP) | OTP email delivery |
| Middlewar e | express-rate-limit | API rate limiting (100 req / 15 min) |
| State Mgmt | React Context API | Auth, price, and trading contexts |
| Dev Tools | pnpm, Git/GitHub, VS Code | Package management, version control |

IV. DATABASE DESIGN

MongoDB was selected as the primary data store for its flexible document-based schema, native JSON compatibility with the Node.js application layer, and support for horizontal scaling. Eight collections are defined, each serving a distinct data concern within the platform.

Table III: MongoDB Collection Definitions

| Collection | Description |
|---------------|--|
| users | Credentials, profile, virtualCash, lockout state |
| otps | OTP value, email, 10-min TTL, used flag |
| refreshTokens | Active refresh tokens with expiry |
| holdings | Per-user equity holdings: symbol, qty, avgBuyPrice |
| transactions | All buy/sell records with price, qty, timestamp |
| watchlist | User-specific stock symbol lists |
| positions | Open F&O positions: contract type, strike, margin |
| stocks | Simulated stock metadata and price history |

Performance indexes are defined on user_id across all user-bound collections and on created_at descending for the transactions collection. Connection pooling with a default pool size of five ensures efficient resource utilisation. The User schema stores virtual cash (default Rs.1,00,000), bcrypt-hashed password, OTP fields, and login attempt tracking. The OTP schema enforces single-use semantics via a used boolean and MongoDB TTL indexes on expiresAt, ensuring automatic document expiration after ten minutes.

Table IV: Next.js Frontend Page Structure

| Route | Description |
|-----------------|--------------------------------------|
| / | Landing page - feature overview |
| /login | Login with OTP verification |
| /signup | Registration form |
| /dashboard | Main trading dashboard |
| /watchlist | Personalised watchlist |
| /holdings | Holdings with P&L + allocation chart |
| /stock/[symbol] | Candlestick chart + order placement |
| /fno | Futures & Options module |
| /positions | Open F&O positions |
| /funds | Virtual funds management |
| /profile | Profile settings |

V. SECURITY ANALYSIS

Luminex implements a six-layer defence-in-depth security model, described in Table V. Each layer addresses a distinct threat class and is independently testable.

Table V: Security Layer Analysis

| Layer | Mechanism | Protects Against |
|------------------|-------------------------------------|-----------------------------|
| Password Storage | bcrypt (saltRounds=10) | Rainbow table / brute-force |
| Session Auth | JWT HMAC-SHA256, HTTP-only cookies | Session hijacking / CSRF |
| OTP Verification | 6-digit, 10-min TTL, single-use | Account takeover |
| API Protection | express-rate-limit (100 req/15 min) | DDoS / brute-force |
| Account Lockout | 5 failures -> 30-min lockout | Credential stuffing |
| Token Rotation | 15-min access + 7-day refresh cycle | Token replay attacks |

All data transmission is enforced over HTTPS in production. OTP documents are invalidated on first use and automatically expired by MongoDB's TTL mechanism. Server-side portfolio calculations prevent exposure of raw pricing algorithms to the client. The stateless JWT architecture supports horizontal scaling without sticky sessions or shared session stores.

VI. RESULTS AND DISCUSSION

A. Functional Testing

Luminex was evaluated across four testing categories: unit, integration, system, and security. All 24 test cases across all categories passed, validating the correctness and robustness of the implemented system.

Table VI: Selected Unit Test Results

| Test Case | Expected Result | Status |
|-----------------------------|---------------------------------|--------|
| Valid OTP within 10 min | JWT issued, verified | Pass |
| Expired OTP entry | Error: OTP expired | Pass |
| 5 failed login attempts | Account locked 30 min | Pass |
| Buy with sufficient funds | Holdings updated, cash deducted | Pass |
| Buy exceeding balance | Error: Insufficient funds | Pass |
| Sell exceeding holdings qty | Error: Insufficient holdings | Pass |
| Tampered JWT | 401 Unauthorised | Pass |
| 101 API requests / 15 min | 429 Too Many Requests | Pass |
| Reuse of consumed OTP | Error: OTP already used | Pass |

B. Performance Metrics

Table VII: Platform Performance Summary

| Metric | Value |
|------------------------------|-------------------|
| API Average Response Time | < 150 ms |
| Stock Price Refresh Interval | 3 seconds |
| OTP Delivery Time (SMTP) | < 5 seconds |
| JWT Access Token Expiry | 15 minutes |
| JWT Refresh Token Expiry | 7 days |
| OTP Expiry | 10 minutes |
| Account Lockout Threshold | 5 failed attempts |
| Lockout Duration | 30 minutes |
| Total Lines of Code | 9,000+ |
| Reusable Components | 30+ |

C. Architecture Discussion

The 3-second price refresh cycle uses efficient client-side polling rather than WebSocket connections. This reduces infrastructure complexity considerably while still providing a near-real-time trading experience. Server-side P&L calculation ensures that raw pricing algorithms are never exposed to the client. MongoDB connection pooling with a default pool size of five ensures efficient database utilisation under concurrent load while remaining configurable for production scaling.

The React Context API is used for three orthogonal state domains: authentication state (auth-context), live price state (price-context), and trading/portfolio state (trading-context). This separation prevents cross-domain state coupling. TypeScript strict mode is applied across the entire frontend codebase, catching type errors at compile time. Next.js 15 with the App Router provides server-side rendering for dashboard and portfolio pages, and static generation for the landing page.

VII. ADVANTAGES AND LIMITATIONS

Advantages

Risk-free learning: Virtual Rs.1,00,000 starting balance with no real money involved. Comprehensive feature set covering equities and F&O derivatives in a single platform. Production-grade security through a multi-layer defence suitable as a real-world reference implementation. Real-time simulation via a 3-second refresh without paid data APIs. Modern industry stack (Next.js 15, TypeScript, Node.js, MongoDB, JWT) directly applicable to industry roles. Zero deployment cost via MongoDB Atlas and Vercel free plans. Dual educational value: teaches both financial market

mechanics and secure full-stack architecture simultaneously.

Limitations

No live exchange integration: Synthetic prices rather than NSE/BSE live feeds. Browser-only with no native mobile application. Single-device sessions with no concurrent multi-device management. No portfolio benchmarking against Nifty 50 or Sensex indices. Email-only OTP with SMS-based OTP not implemented. Limited load testing: concurrent user performance under thousands of simultaneous users has not been benchmarked.

VIII. FUTURE SCOPE

Several directions are identified for future enhancement. Live market data integration with NSE/BSE APIs or third-party providers such as Alpha Vantage or Fyers would replace synthetic price simulation with real market feeds. AI-based stock recommendations using LSTM or ARIMA models trained on historical price data would provide personalised suggestions based on individual user trading patterns. A React Native mobile application would extend platform reach to Android and iOS. Cloud deployment pipelines on AWS or GCP with GitHub Actions CI/CD would enable automated testing and deployment.

Advanced analytics including Sharpe ratio, maximum drawdown, and portfolio beta computation would deepen the financial education utility. Competitive leaderboard and social trading features would gamify learning without sacrificing simulation realism. SMS OTP via Twilio or MSG91 would provide an alternative authentication channel. An algorithmic trading module would allow users to write and back-test simple rule-based strategies using predefined templates.

IX. CONCLUSION

Luminex addresses a real and persistent gap: the absence of a single, freely available, secure, and feature-complete virtual trading platform that can serve simultaneously as a financial education tool and as a production-grade reference implementation for secure full-stack web development. By integrating OTP-based authentication, JWT session management, bcrypt password hashing, rate limiting, and account lockout protection with a realistic equity and F&O trading simulation built on Next.js 15, TypeScript, Node.js, Express.js, and MongoDB, the platform demonstrates that educational utility and production-grade engineering quality are not competing objectives - they are mutually reinforcing.

The successful completion of all unit, integration, system, and security test cases, combined with API response times under 150 ms and a 3-second price refresh cycle, validates both the correctness and the practical usability of the system. Future integration of live market data, AI-based recommendations, and mobile application support will further elevate Luminex from an educational simulator to a production-ready fintech platform.

The technology stack - Next.js, TypeScript, Node.js, Express.js, and MongoDB - directly mirrors current industry practice, ensuring that students who build with and on Luminex are acquiring skills with immediate professional applicability.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Dr. Shubhangi R. Patil for her invaluable guidance, continuous encouragement, and expert supervision throughout the development of this project. The authors also thank the Department of Computer Engineering, Sinhgad Institute of Technology, Lonavala, for providing the necessary resources and support.

REFERENCES

- [1] NSE India, "NSE Paathshaala - Investor Education," National Stock Exchange of India, 2024. [Online]. Available: <https://www.nseindia.com/invest/paathshaala>
- [2] A. Sharma, "Secure Authentication in Financial Systems," IEEE Access, vol. 11, 2023.

- [3] OWASP Foundation, "OWASP Top Ten 2023," 2023. [Online]. Available: <https://owasp.org/www-project-top-ten/>
- [4] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," RFC 7519, IETF, May 2015. [Online]. Available: <https://tools.ietf.org/html/rfc7519>
- [5] J. Brown, "Educational Trading Simulators and Financial Literacy," IEEE Transactions on Education, vol. 65, no. 3, 2022.
- [6] N. Provos and D. Mazieres, "A Future-Adaptable Password Scheme," USENIX Annual Technical Conference, 1999.
- [7] MongoDB, Inc., "MongoDB Documentation," 2024. [Online]. Available: <https://www.mongodb.com/docs/>
- [8] Vercel, Inc., "Next.js 15 Documentation," 2024. [Online]. Available: <https://nextjs.org/docs>
- [9] OpenJS Foundation, "Express.js Documentation," 2024. [Online]. Available: <https://expressjs.com/>
- [10] J. C. Hull, Options, Futures, and Other Derivatives, 10th ed., Pearson Education, 2017.
- [11] Tailwind Labs, "Tailwind CSS Documentation," 2024. [Online]. Available: <https://tailwindcss.com/docs>
- [12] Microsoft, "TypeScript Documentation," 2024. [Online]. Available: <https://www.typescriptlang.org/docs/>