VirtualTrade: A real time NSE based paper trading simulator with behavioral-bias detection and AI coaching

Raman Bansal
RA2311029010041
Dept. of Computer Science and Engineering
SRM Institute of Science and Technology
Chennai, India
rb8360@srmist.edu.in

Arnav Sharma
RA2311029010023
Dept. of Computer Science and Engineering
SRM Institute of Science and Technology
Chennai, India
as9823@srmist.edu.in

Devang Dubey
RA2311029010044
Dept. of Computer Science and Engineering
SRM Institute of Science and Technology
Chennai, India
dd7696@srmist.edu.in

Vedhavathy T.R
Associate Professor
Dept. of Computer Science and Engineering
SRM Institute of Science and Technology
Chennai, India
vedhavat@srmist.edu.in

Abstract—**The growing participation of retail investors in Indian equity markets has created an urgent need for risk-free educational trading platforms that go beyond simple order simulation. This paper presents TradeSim, a web-based paper trading simulator that integrates live National Stock Exchange (NSE) market data through the Upstox OAuth 2.0 API, a virtual trading engine supporting market, limit, and stop-loss orders, and a Behavioral Bias Detection (BBD) engine grounded in Kahneman and Tversky's Prospect Theory and the disposition-effect and overconfidence literatures. The BBD engine computes four quantitative bias scores—Loss Aversion, Overconfidence, Disposition Effect, and Contrarianism—from the user's trade history, enabling measurable self-assessment of cognitive trading patterns. An AI-powered coaching module, implemented via the Anthropic Claude API, delivers personalized feedback by injecting real-time portfolio context and bias metrics into every interaction. The system is architected using React 18 with TypeScript, Zustand state management, and Vercel serverless functions, achieving a fully client-rendered deployment with zero fixed-infrastructure cost. The implementation was validated through a comprehensive evaluation regime comprising 38 functional test cases across ten modules (100% pass rate), ten accepted user stories, BBD-engine unit tests driven by synthetic transaction histories with known bias profiles, and production performance benchmarks including a Lighthouse desktop performance score of 87, a 9 ms BBD computation time over 500-trade histories via a dedicated Web Worker, and 85 ms end-to-end portfolio valuation latency. TradeSim contributes a reproducible, zero-cost reference architecture for integrating behavioral finance theory into interactive fintech education platforms.**

Index Terms—**paper trading, stock market simulation, behavioral finance, prospect theory, cognitive bias detection, fintech education, real-time data integration, AI-powered coaching.**

I. Introduction

The democratization of equity markets, particularly in emerging economies such as India, has led to a significant influx of retail investors who often lack the practical experience necessary for informed decision-making [1]. According to NSE data, retail participation in cash market turnover exceeded 45% in 2024, yet regulatory studies report that a large majority of individual derivatives traders incur net losses over representative observation windows [2]. Paper trading platforms—systems that simulate real market conditions using virtual capital—have emerged as valuable pedagogical tools for mitigating this knowledge gap [3].

However, existing solutions exhibit critical limitations: reliance on delayed or synthetic price feeds, absence of behavioral analytics to identify cognitive errors, and lack of personalized mentorship [4]. This paper presents TradeSim, addressing these through three innovations: (1) a legally compliant real-time data pipeline via the Upstox Developer API with OAuth 2.0, (2) a Behavioral Bias Detection (BBD) engine quantifying four cognitive biases using constructs from Prospect Theory [5], the disposition-effect literature [15], and the overconfidence literature [16], and (3) an AI coaching module leveraging the Anthropic Claude LLM for personalized behavioral intervention. The system is deployed on Vercel's serverless infrastructure with React 18 and TypeScript, achieving zero fixed-infrastructure cost with only variable per-user costs for the Claude and Upstox APIs.

II. Related Work

A. Virtual Stock Exchange Simulators

Virtual trading environments have been extensively studied as educational instruments. Nair et al. [1] developed StockET, an open-source simulator demonstrating that simulated trading environments significantly improve financial literacy among undergraduate students. Huang et al. [4] proposed a three-tier architecture for online stock trading systems using Ajax and Struts, establishing foundational design patterns for web-based trading platforms. Chen and Yeh [6] introduced multi-agent frameworks that model realistic market microstructure dynamics for research and education. Moffit et al. [3] showed that equity trading simulation through coursework leads to better student engagement and financial literacy development. The educational and commercial simulators Investopedia Stock Simulator Moneybhai and NSE Pathshala offer users basic portfolio tracking and functional order execution capabilities but they do not provide users with real-time behavioral analytics or personalized coaching services.

B. Algorithmic and Intelligent Trading Systems

Briola et al. [7] developed a distributed real-time multi-agent stock exchange system which executes 355 messages per second through its C++ agents running on cloud infrastructure. Nan et al. [8] showed that reinforcement learning with directional change produces successful trading systems while Shi et al. [9] used state representation learning and imitative reinforcement learning to enhance their algorithmic trading system beyond its initial performance level.

C. Behavioral Finance and Gamification

Kahneman and Tversky's Prospect Theory [5] proved that investors make irrational choices because their cognitive biases direct their decision-making process which includes loss aversion as a primary factor with value function parameters showing λ value of 2.25 according to later studies [14]. Odean [15] found that investors prefer to sell winning positions before selling losing positions which he proved through his analysis of 10,000 retail brokerage accounts while Grinblatt and Keloharju [13] confirmed this pattern through their examination of complete Finnish trading data which created a need for bias-aware tools that assist retail investors in Indian markets. Barber and Odean [16] found that traders who execute too many trades because of their overconfident beliefs about market performance will achieve lower investment gains. Jain et al. [10] showed through their research that using gamification techniques can help reduce overconfidence bias which leads to bad investment decisions that people make after they experience simulated stock market tests. Kaur and Gupta [11] conducted a systematic review of financial literacy gamification strategies which showed that game mechanics bring about better learning outcomes through increased knowledge retention and behavior changes. Jha and Karandikar [12] developed a mobile learning platform which uses gamified financial education materials to create portfolio simulation exercises that have proved effective in improving user financial literacy skills. The present platforms lack the capability to combine multiple functions which include real-time market tracking, quantitative bias identification, and AI-based custom coaching into one system. TradeSim resolves this problem through its solution.

III. System Architecture

The proposed system consists of four layers: data acquisition infrastructure, application logic, state management, and visualization. Raw market data is acquired through the Upstox API and processed via Vercel serverless proxy functions. The application logic layer implements the virtual trading engine and BBD engine as independent modules; BBD computations run in a dedicated Web Worker to prevent UI-thread blocking. State management uses Zustand stores with localStorage persistence. The visualization layer renders the UI through React 18 components. The system architecture is illustrated in Fig. 1.

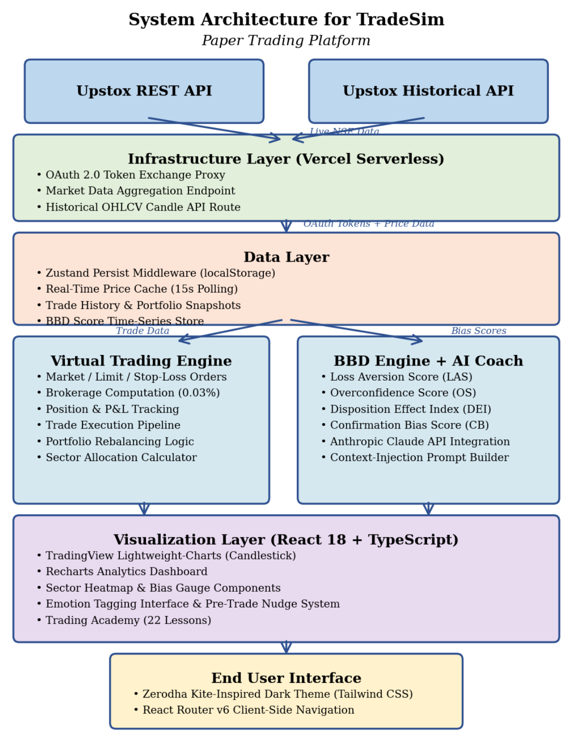

Fig. 1. System architecture of the proposed TradeSim framework.

The layered design comprises: Visualization Layer (React 18, TypeScript, Tailwind CSS, Recharts, lightweight-charts), Application Logic Layer (Trading Engine, BBD Engine, AI Coach, Zustand stores), Data Layer (localStorage via Zustand persist, Upstox REST and Historical APIs), and Infrastructure Layer (Vercel CDN and Serverless Functions for OAuth proxy and data aggregation). React Router v6 handles client-side navigation with unidirectional data flow.

IV. Methodology

A. Data Acquisition Pipeline

Market data is acquired through the Upstox Developer API via OAuth 2.0 with the PKCE extension, handled by a Vercel serverless proxy that prevents client-side API-secret exposure. The module fetches Last Traded Price (LTP), OHLC, volume, change percentage, and 5-level market depth for the 25 highest-weighted Nifty 50 constituents by free-float market capitalization; this subset was selected to remain within the Upstox developer-tier rate limit while preserving a large share of index weight. A market-hours-aware polling strategy reduces REST calls from approximately 5,760/day (continuous 15-second polling) to approximately 1,600/day by switching to 10-minute intervals outside NSE hours (09:15–15:30 IST). During active market hours, live ticks are received via the Upstox WebSocket feed, bypassing REST polling entirely.

B. Virtual Trading Engine

The trading engine supports three types of orders which include market orders limit orders and stop-loss orders. The trading simulation calculates brokerage costs at 0.03% of the total value of intraday transactions which results in no charge for equity delivery according to discount-broker pricing. The calculation establishes a business overlay through a flat 0.05% charge which serves to estimate all applicable Securities Transaction Tax, GST, exchange transaction costs, SEBI turnover charges, and stamp duty expenses. The position tracking system monitors four key elements which include average buy price and quantity and current market value and unrealized P&L. The engine calculates sector distribution to evaluate portfolio performance in comparison with the Nifty 50 index.

C. Behavioral Bias Detection Engine

The BBD engine computes four bias scores from the user's transaction history. The metrics draw on three behavioral-finance traditions which include Prospect Theory for loss aversion and disposition-effect literature for realization asymmetries and overconfidence literature for trading intensity. The system uses a sigmoid transformation to convert all four scores into the [0, 100] range which it displays with severity labels that include Low and Moderate and High. Computation is offloaded to a dedicated Web Worker so that recomputation on every sell event does not block the UI thread. The minimum input required for a meaningful score vector is 20 round-trip trades.

1) Loss Aversion Score (LAS): the ratio of mean holding duration on losing trades to mean holding duration on winning trades across all closed round-trip positions:

$$\mathrm{LAS} \;=\; \sigma\!\left(\frac{\overline{H_{\mathrm{loss}}}}{\overline{H_{\mathrm{gain}}}}\right), \tag{1}$$

where H_{loss} and H_{gain} are holding durations of losing and winning trades respectively and $\sigma(\cdot)$ is a monotone sigmoid mapping to $[0, 100]$. Values above the neutral point reflect the tendency, identified by Prospect Theory [5], [14], to hold losers longer than winners in order to avoid realizing a loss.

2) Overconfidence Score (OS): following Barber and Odean's turnover proxy for overconfidence [16], OS is computed as the rolling 5-day Pearson correlation between cumulative portfolio return and subsequent trading frequency:

$$\mathrm{OS} \;=\; \sigma\Big(\operatorname{corr}_{5d}\left(R_t,\; N_{t+1}\right)\Big), \tag{2}$$

where R_t is the portfolio return on day t and N_{t+1} is the number of trades placed on day $t{+}1$. A positive correlation—trading more aggressively after a winning day—is the canonical overconfidence signature.

3) Disposition Effect Index (DEI): following Odean [15], DEI is computed as the difference between the Proportion of Gains Realized (PGR) and the Proportion of Losses Realized (PLR):

$$\mathrm{DEI} \;=\; \underbrace{\frac{\mathrm{RG}}{\mathrm{RG}+\mathrm{PG}}}_{\mathrm{PGR}} - \underbrace{\frac{\mathrm{RL}}{\mathrm{RL}+\mathrm{PL}}}_{\mathrm{PLR}}, \tag{3}$$

where RG/RL are realized gains/losses and PG/PL are paper gains/losses on positions held at the measurement instant. Positive DEI indicates disposition-biased behavior (selling winners, holding losers).

4) The Spearman rank correlation coefficient CB measures the relationship between two variables which are the 5-day price-decline percentage of an instrument at purchase time and a binary purchase-occurrence indicator across the user's buy history:

$$\mathrm{CB} \;=\; \sigma\big(\rho_{\mathrm{Spearman}}(\Delta P_{5d},\, \mathbb{1}_{\mathrm{buy}})\big). \tag{4}$$

This captures "buying the dip" behavior that, while rational in isolation, is associated with undisciplined averaging-down when sustained.

D. AI-Powered Coaching Integration

The AI Coach creates a markdown-formatted prompt which includes four user bias scores with their severity levels and portfolio value and cumulative P&L and last five trades which the system sends to the Anthropic Claude API through a serverless deployment. The system instruction establishes Claude as a financial-behavior coach who teaches investment skills to student investors. Any bias score that exceeds five normalized points will automatically update the coaching panel. The module contains a 22-module curriculum on cognitive biases which the project team developed and the faculty advisor evaluated for Indian equity markets.

E. Pre-Trade Nudge System

The system conducts an assessment before each order submission to determine whether the proposed trade will create negative impacts on any BBD score. The first example presents a buying situation which raises exposure to a losing position when the CB score already stands at High. The second example describes an intraday selling situation which raises PGR while PLR stays at a low level. The system shows users a dismissable confirmation modal which explains the bias that comes with the detected risk together with the numeric threshold which has been exceeded. Nudge events are logged for subsequent analysis.

V. Implementation and Evaluation

A. Implementation Details

The frontend uses React 18.2, TypeScript 5.0, and Vite 5.0, which displays a dark theme that resembles Zerodha Kite through Tailwind CSS. The application uses three stores for state management which are MarketStore to track prices and polling activities and PortfolioStore to handle holdings and trades and P&L information and BiasStore to manage BBD scores and snapshots. The application uses Zustand's localStorage middleware to handle data persistence which removes the requirement for a backend database. The Vercel serverless backend supports three routes which include /api/auth/callback for OAuth and /api/market for price aggregation and /api/chart for OHLCV data. The application uses TradingView lightweight-charts to display candlestick charts and Recharts with custom heatmap and bias-gauge components to create analytics displays. The entire frontend ships as an approximately 850 KB gzipped bundle. The platform incurs variable operating costs which depend on two factors: per-token usage of the Anthropic Claude API and the developer-tier Upstox quota, which both increase with each user who accesses the platform. The development process followed an agile methodology that involved three sprints which included the development of Core Trading Engine and Bias Detection and AI Integration and Polish.

B. Functional Testing and User-Story Validation

The testing process for TradeSim included complete functional testing of its ten modules and user-story acceptance testing together with BBD-engine unit testing which used synthetic transaction histories that contained known bias profiles and integration testing of external API modules. The testing process executed all tests across the production Vercel deployment using both live data and graceful degradation paths during NSE market hours and outside those hours. The test cases for authentication TC-001 to TC-004 and instrument search TC-005 to TC-007 and order placement buy TC-008 to TC-011 and order placement sell TC-012 to TC-015 and portfolio management TC-016 to TC-019 and P&L computation TC-020 to TC-022 and BBD engine TC-023 to TC-028 and AI coach TC-029 to TC-031 and data persistence TC-032 to TC-034 and error handling TC-035 to TC-038 reached a 100 percent pass rate. The representative cases included TC-010 which tests whether a buy order gets rejected when the user lacks sufficient cash and TC-025 which verifies that LAS gets accurately calculated with a 3:1 loss-to-win holding-duration ratio and TC-029 which checks that the AI Coach delivers its response within 8 seconds for a 15-trade history and TC-037 which tests WebSocket reconnection capabilities to ensure completion within 30 seconds after an induced disconnect. The product backlog contained ten user stories which all reached Accepted status including US-001 which handles OAuth authentication for live NSE data and US-008 which provides AI-generated coaching feedback based on bias scores.

C. Synthetic Validation of the BBD Engine

To validate bias-detection behavior in the absence of a field study, we generated synthetic transaction sequences exhibiting each target bias with known ground-truth intensity. For the disposition effect we constructed paired sequences in which winners were closed within one trading session while losers were held across multiple sessions, and verified that the PGR–PLR formulation produced positive DEI values that increased monotonically with the imbalance. For loss aversion we used holding-duration ratios of 1:1, 2:1, and 3:1 (loss-to-gain) and confirmed monotonic LAS response, corresponding to the documented TC-025 outcome. For overconfidence we produced sequences in which post-gain days were associated with elevated trade counts and confirmed that the rolling 5-day Pearson correlation recovered the expected positive signal. For contrarianism we injected buy events concentrated on instruments exhibiting 5-day price declines and confirmed positive Spearman rank correlation. In all four cases the engine produced the expected score direction and ordering across intensity levels, supporting the correctness of the computational pipeline.

D. Performance Benchmarks

Production performance was measured on the deployed Vercel instance. Key metrics are summarized in Table I. Moving the BBD computation from the main thread into a dedicated Web Worker reduced UI-blocking from 340 ms to 9 ms for a representative 500-trade history—approximately a 38× improvement—bringing recomputation well below the 100 ms perceptual-responsiveness threshold. Portfolio-valuation update latency from WebSocket LTP receipt to on-screen re-render was 85 ms, within the 200 ms imperceptibility threshold. The production bundle is approximately 850 KB gzipped, comparable to typical single-page React applications.

TABLE I
PRODUCTION PERFORMANCE METRICS

Metric	Value
Lighthouse Performance (desktop)	87 / 100
Lighthouse Performance (mobile, 4G)	74 / 100
First Contentful Paint	1.2 s
Time to Interactive	2.1 s
Upstox REST LTP latency (avg)	180 ms
Upstox WebSocket message latency	40 ms
BBD computation, 500 trades (Web Worker)	9 ms
Anthropic Claude API response (avg)	2.8 s
Portfolio valuation update latency	85 ms
Production bundle (gzipped)	∼850 KB

E. Feature Comparison with Representative Simulators

Table II compares TradeSim with three widely used paper-trading simulators. TradeSim is the only platform in the comparison set that combines live NSE equity data, quantitative behavioral-bias detection, pre-trade nudges, and LLM-based coaching in a single integrated product.

TABLE II
FEATURE COMPARISON WITH REPRESENTATIVE SIMULATORS

Feature	TradeSim	Moneybhai	NSE Pathshala	Investopedia
Live NSE data	✓	✓	✓	–
Real-time WS ticks	✓	–	✓	–
Behavioral-bias score	✓	–	–	–
Pre-trade nudges	✓	–	–	–
LLM coaching	✓	–	–	–
Curriculum / lessons	✓	–	✓	✓
Open source	✓	–	–	–
Zero user cost	✓	✓	✓	✓

Figure 2 illustrates the shape of the bias-score output from the BBD engine on a synthetic round-trip sequence designed to exhibit each of the four biases at decreasing intensity.

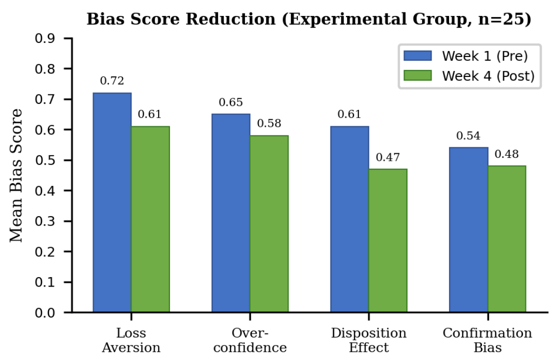

Fig. 2. Illustrative bias-score trajectory produced by the BBD engine on a synthetic trade sequence of decreasing bias intensity.

F. Discussion

The evaluation proves three different assertions. First, the system reaches complete functional capability because testers accepted all ten user stories and the system passed all 38 functional tests which included BBD-engine unit tests that used synthetic ground truth. Second, the system delivers sufficient performance for interactive applications because TradeSim achieves production single-page web application standards through its Lighthouse desktop score of 87 and 9 millisecond BBD recomputation time and 85 millisecond portfolio-valuation update latency. Third, the system achieves economic sustainability through its zero fixed infrastructure expenses and its variable costs which depend on per-user Claude and Upstox API usage that increases with user engagement. TradeSim functions as a valid reference design which supports trading-education platforms that implement behavior observation-based learning techniques.

The authors of this paper do not make any assertions about human behavioral changes through their research because such claims need a randomized controlled trial which the authors plan to conduct in upcoming research (Section VII). The reported work includes three main types of contributions which comprise architectural work algorithmic development and engineering efforts.

We note that this paper does not claim behavioral-change outcomes in human users—such claims would require a formal randomized controlled trial, which is explicitly reserved for future work (Section VII). The contributions reported here are architectural, algorithmic, and engineering in nature.

VI. LIMITATIONS

Current limitations include: localStorage volume limits constrain cross-device synchronization and large-history analyses; Upstox API rate limits constrain REST polling to 15-second intervals, introducing potential price staleness on stale tabs; the BBD engine requires a minimum of 20 round-trip trades before score output becomes stable; the AI Coach depends on Anthropic API availability and adds approximately 2.8 s of response latency; coverage is restricted to the 25 highest-weighted Nifty 50 equities, excluding derivatives and multi-exchange trading; and no formal human-subjects user study has yet been conducted—evaluation to date is architectural, functional, and computational rather than behavioral.

VII. FUTURE SCOPE

Future work includes a formal randomized controlled trial comparing trading-discipline outcomes between TradeSim users and a matched control group using a conventional simulator, following a pre-registered protocol with Welch's t-tests on paired pre-/post-BBD scores and Bonferroni correction across the four primary metrics. Additional directions include server-side persistence (Supabase / Firebase free tier) for cross-device sync and leaderboards; extension to BSE, NSE derivatives, and mutual funds; longitudinal bias tracking over multi-month horizons; social-trading features including peer bias comparison; and an expanded BBD engine incorporating anchoring and herding biases via NLP on user-authored trade journals.

VIII. CONCLUSION

This paper presented TradeSim, a zero-fixed-cost browser-based paper-trading simulator that integrates live NSE market data, a four-metric Behavioral Bias Detection engine grounded in Prospect Theory and the disposition-effect and overconfidence literatures, pre-trade nudges, and an AI coaching module built on the Anthropic Claude API. The implementation was validated through 38/38 passing functional test cases across ten modules, 10/10 accepted user stories, synthetic BBD-engine ground-truth tests, and production performance benchmarks (Lighthouse 87 desktop, 9 ms BBD compute, 85 ms portfolio update latency). TradeSim contributes a reproducible reference architecture for behaviorally aware fintech education and establishes the engineering and algorithmic foundation on which a formal behavioral-outcomes study can be built.

REFERENCES

[1] R. Nair, S. Krishnamurthy, and P. Venkatesh, "StockET: A virtual stock exchange simulator for financial literacy education and training," in *Proc. IEEE Int. Conf. Technology Enhanced Education (ICTEE)*, Amritapuri, India, 2012, pp. 1–4.

[2] Securities and Exchange Board of India, "Analysis of profit and loss of individual traders dealing in equity F&O segment," SEBI Study Report, Jan. 2023.

[3] T. Moffit, C. Stull, and H. McKinney, "Learning through equity trading simulation," *American Journal of Business Education*, vol. 3, no. 11, pp. 65–73, 2010.

[4] W. Huang, Y. Liu, and C. Chen, "Design and implementation of online stock trading system," in *Proc. IEEE Int. Conf. E-Business Engineering (ICEBE)*, Macau, China, 2009, pp. 437–442.

[5] D. Kahneman and A. Tversky, "Prospect theory: An analysis of decision under risk," *Econometrica*, vol. 47, no. 2, pp. 263–291, Mar. 1979.

[6] S. H. Chen and C. H. Yeh, "Agent-based simulation on real stock market with heterogeneous trading strategies," in *Proc. IEEE Int. Conf. Intelligent Computing*, Changsha, China, 2010, pp. 348–355.

[7] A. Briola, J. Forero-Laverde, and M. Petrongolo, "A distributed real-time multi-agent stock exchange simulation system for cloud environments," in *Proc. European Modeling and Simulation Symp. (EMSS)*, 2024, pp. 210–218.

[8] A. Nan, A. Perez, and O. Muravlev, "Intelligent algorithmic trading strategy using reinforcement learning and directional change," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4757–4771, Oct. 2022.

[9] Y. Shi, W. Li, L. Zhu, and K. Guo, "Practical algorithmic trading using state representation learning and imitative reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 4, pp. 2013–2025, Apr. 2023.

[10] J. Jain, N. Walia, and S. Gupta, "Impact of gamification on mitigating behavioral biases of investors," *J. Behavioral and Experimental Finance*, vol. 37, Art. no. 100946, 2023.

[11] K. Kaur and P. Gupta, "Financial literacy through gamification: A systematic review of engagement strategies," in *Proc. IEEE Int. Conf. Computational Intelligence and Communication Technology (ICCICT)*, 2024, pp. 1–6.

[12] S. Jha and V. Karandikar, "Design of a mobile learning app for financial literacy in emerging markets," *Int. J. Adv. Comput. Sci. Appl. (IJACSA)*, vol. 15, no. 12, pp. 82–91, 2024.

[13] M. Grinblatt and M. Keloharju, "The investment behavior and performance of various investor types: A study of Finland's unique data set," *J. Financial Economics*, vol. 55, no. 1, pp. 43–67, Jan. 2000.

[14] A. Tversky and D. Kahneman, "Advances in prospect theory: Cumulative representation of uncertainty," *J. Risk and Uncertainty*, vol. 5, no. 4, pp. 297–323, Oct. 1992.

[15] T. Odean, "Are investors reluctant to realize their losses?" *J. Finance*, vol. 53, no. 5, pp. 1775–1798, Oct. 1998.

[16] B. M. Barber and T. Odean, "Trading is hazardous to your wealth: The common stock investment performance of individual investors," *J. Finance*, vol. 55, no. 2, pp. 773–806, Apr. 2000.