iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: V Month of publication: May 2026

DOI: https://doi.org/10.22214/ijraset.2026.81969

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

Couture Connect: An Adaptive AI Agent for Boutique Retail Automation

Pothani Akshaya Bhavani[1], Pravalika Gande[2], Tanvi Kovai[3], Saachi Manish Bhalgat[4]
Department of Computer Science and Engineering Keshav Memorial Institute of Technology Narayanaguda, Hyderabad, India

Abstract- CoutureConnect is an Adaptive AI Agent prototype designed for small boutiques to transform retail engagement and streamline operational workflows by addressing the core challenges of inconsistent digital presence, delayed customer communication, and the absence of structured customer records. The system integrates an Adaptive Content Engine that generates Instagram and WhatsApp-ready social media drafts while continuously learning from past engagement data to refine content strategies, alongside an Agentic Customer Flow that automates personalized customer notifications for order updates and new product launches while maintaining evolving customer profiles containing sizes, style preferences, and spend patterns. By proactively executing tasks based on internal triggers such as order completion and product releases, CoutureConnect delivers measurable improvements in time savings, engagement uplift, and notification coverage, offering a scalable and sustainable growth enabler for boutique retailers operating without dedicated marketing or CRM infrastructure.
Keywords - AI Agent, Boutique Retail, CRM Automation, Adaptive AI, Agentic Workflow, Social Media Automation, Instagram, WhatsApp, Small Business Technology

I. INTRODUCTION

Small boutique retailers face a significant disadvantage in today's digital market. Large retail chains have dedicated marketing teams, enterprise CRM platforms, and automated content tools, whereas a typical boutique owner with one or two staff members is expected to compete on the same platforms without comparable resources. The same person managing inventory and serving walk-in customers is also expected to post consistently, reply promptly, and remember each customer's preferences,which is impractical in day-to-day operations.

Existing tools are not designed around boutique workflows. Standalone content tools generate posts but retain no memory of what performed well previously. Enterprise CRM systems require implementation teams and IT budgets. Messaging platforms often assume a WhatsApp Business API setup that most small owners have never encountered. Each tool solves one part of the problem, and few are integrated in a way that is feasible at boutique scale.

Couture Connect is designed around how boutiques actually work. It combines an Adaptive Content Engine that drafts and improves Instagram and WhatsApp campaigns based on past performance with an Agentic Customer Flow that sends personalized notifications automatically when orders move through their lifecycle or when new products arrive. Both components share a single customer database, so the same profile data that personalizes a notification also shapes the next content draft, providing an integrated alternative to fragmented tools. This pattern reflects a broader shift in service research, where artificial intelligence is recognized as a core driver of new service configurations and performance improvements across customer-facing processes [3].

II. RELATED WORK

Reference	Methodology / Approach	Primary Focus	Key Limitations
Grewal et al. [1]	AI adoption review	Enterprise forecasting	No SME applicability
Verhoef et al. [2]	Digital transformation framework	AI-driven personalization	Assumes infrastructure boutiques lack
Kumar et al. [4]	Conceptual and managerial framework for AI-driven engagement marketing	Role of artificial intelligence in personalized engagement across the customer journey	Focuses on large-firm marketing strategy rather than lightweight tools for boutique retailers

Russell and Norvig [5]	Goal-directed autonomous agents	Foundational agentic AI	No applied retail context
Wooldridge and Jennings [6]	Multi-agent systems theory	Intelligent agent classification	Predates modern LLM implementations
Xu et al. [7]	Rule-based chatbot for social media	Customer support automation	No adaptive content component
Yao et al. [8]	ReAct framework for LLM agents	Multi-step task execution	Too complex for boutique deployment
Li et al. [9]	Contextual bandit for content selection	Engagement-based recommendation	News context only; not fashion retail
Kim and Ko [10]	Social media engagement in fashion	Brand equity and engagement	Static study; no automation
Reinartz et al. [12]	CRM process quality measurement	Retention and profitability	Large-firm context only
Bull [13]	CRM failure in SMEs	Adoption misalignment	No lightweight alternative proposed
Appel et al. [14]	Social media as brand channel	Instagram and WhatsApp in retail	Does not address boutique automation

The existing literature on retail AI spans enterprise-focused transformation studies, foundational agent theory, adaptive content research, CRM adoption analyses, and social media automation work. Enterprise retail research consistently demonstrates that personalization and automation deliver measurable value, but assumes infrastructure and staffing that boutique operators simply do not have. Agentic systems research establishes that autonomous goal-directed behavior is technically feasible, but existing implementations operate at a complexity level impractical for single-owner deployment. Adaptive content work confirms that engagement-based refinement outperforms static strategies in fashion retail specifically, but does not connect content generation to customer data or notification workflows. CRM research identifies boutique adoption barriers without proposing lightweight alternatives. Payne and Frow's strategic CRM framework emphasizes cross-functional processes, integrated information management, and multichannel interaction [11], but assumes organizational structures and resources that single-owner boutiques typically lack.

No existing work integrates adaptive content generation, agentic customer notification, and structured customer profile management within a single architecture designed for the boutique retail context. CoutureConnect addresses this gap directly.

III. METHODOLOGY

A. *Research Design*

This aligns with the design science paradigm in information systems, in which an IT artifact is iteratively designed, implemented, and evaluated to address a relevant organizational problem [15]. The research moves through three phases: problem identification, system development, and prototype evaluation.

B. *System Materials*

- Frontend: React.js, Vite, Tailwind CSS, Recharts
- Backend: FastAPI, Python
- Database: Supabase (PostgreSQL)
- AI Components: Adaptive Content Generation Engine, Rule-based fallback generator

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

- Dev Tools: Git, GitHub, Node.js, Visual Studio Code

C. System Development Method

An Agile methodology was followed across a four-week development cycle, with four modules developed in parallel by separate team members. A shared Supabase schema, consistent UUID format, and standardized API naming conventions ensured clean integration across independently developed modules. Schema changes were frozen after the first development day to preserve cross-module data consistency.

D. Data Collection and Management

Customer data is stored in Supabase and includes personal identifiers, purchase history, style preferences, size profiles, and engagement patterns. Profiles are updated continuously as order events are processed and campaign interactions are recorded. A simulated dataset of 58 customers, 147 orders, and five weeks of operational activity was constructed to support prototype evaluation.

E. Customer Segmentation — RFM Model

Customers are segmented into three groups using a simplified RFM model covering recency, frequency, and monetary value. Each dimension is scored on a normalized scale and combined as follows:

$$RFM_{Score} = w_1 \cdot R + w_2 \cdot F + w_3 \cdot M$$

Where R = recency score (days since last purchase, inverted), F = frequency score (number of orders in the evaluation window), M = monetary score (cumulative spend), and w_1, w_2, w_3 are weights set to 0.30, 0.30, and 0.40 respectively, giving slightly higher weight to spend as a retention signal. Customers are then classified as:

VIP: RFM Score ≥ 0.70
Regular: RFM Score (0.40 - 0.69)
At-Risk: RFM Score < 0.40

F. Customer Lifetime Value (CLV) Scoring

The CRM module computes a simplified CLV score for each customer to support prioritization in the analytics dashboard:

$$CLV = AOV \times PF \times L$$

Where AOV = Average Order Value (total spend / number of orders), PF = Purchase Frequency (orders per week in the evaluation window), and L = estimated Customer Lifespan in weeks derived from the customer's earliest recorded order date. This formula provides a lightweight CLV estimate appropriate for the small transaction volumes typical of boutique retail.

G. Engagement Rate Calculation

Campaign performance is measured using the standard engagement rate formula applied to simulated interaction data:

$$Engagement\ Rate\ (\%) = \frac{Likes + Comments + Shares}{Total\ Reach} \times 100$$

This metric is computed after each campaign iteration and stored in Supabase. The adaptive content engine retrieves these values when constructing prompts for subsequent generations, using them to identify which content attributes correlate with higher engagement.

H. Notification Coverage Rate

The agentic workflow's effectiveness is measured as:

This metric is computed separately for each trigger type the order placed, order shipped, order delivered, VIP product alert, and at-risk re-engagement and in aggregate across all 140 trigger events in the evaluation window.

$$Coverage\ Rate\ (\%) = \frac{Notifications\ Dispatched}{Total\ Trigger\ Events} \times 100$$

I. Time Savings Estimation

Weekly time savings are estimated by comparing pre-deployment manual effort, derived from the boutique owner's time logs, against measured automated effort in the prototype:

$$Time\ Savings\ (\%) = \frac{Manual\ Effort - Automated\ Effort}{Manual\ Effort} \times 100$$

Manual effort estimates are based on the boutique owner's self-reported time logs over four weeks of pre-deployment operation, which indicated that content creation, order follow-ups, and customer record maintenance together required approximately eight to twelve hours per week.Automated effort is measured as actual time spent on content approval and campaign review in the prototype evaluation.

J. Adaptive Uplift Calculation

Campaign improvement across iterations is measured as engagement uplift relative to the baseline iteration:

$$Uplift\ (\%) = \frac{Engagement_i - Engagement_1}{Engagement_1} \times 100$$

Where $Engagement_i$ is the engagement rate at iteration i and $Engagement_1$ is the baseline engagement rate at iteration 1 using generic template content.

K. Adaptive Content Generation Method

Content generation follows a four-step pipeline. Input parameters including product type, target platform, and customer segment are received. A prompt is constructed using these inputs together with stored engagement history from prior campaigns. Content is generated via AI model or rule-based template depending on availability. Output is stored in Supabase and tagged with engagement metadata for use in future generation cycles. Over successive iterations, the engine adjusts tone, format, and structural choices based on which content attributes correlated with higher engagement in prior rounds.

L. Agentic Workflow Method

The system autonomously executes notification workflows based on internal triggers without requiring manual input:

The system monitors trigger conditions including order status changes, new product additions, and inventory threshold crossings

Relevant customer segments are identified based on trigger type and stored profile data

Personalized messages are generated using customer name, order details, and style preference data

Notifications are dispatched through the API layer and logged in Supabase for audit and reporting

M. Testing and Validation

The system was tested at four levels:

- Unit testing: Individual functions and agent trigger logic
- Integration testing: Data flow across all four modules via the shared Supabase layer
- Functional testing: End-to-end workflow execution from trigger detection to notification dispatch
- User testing: Dashboard usability and content approval flow

Postman was used for API validation and standard Python testing frameworks for backend verification.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

IV. SYSTEM ARCHITECTURE

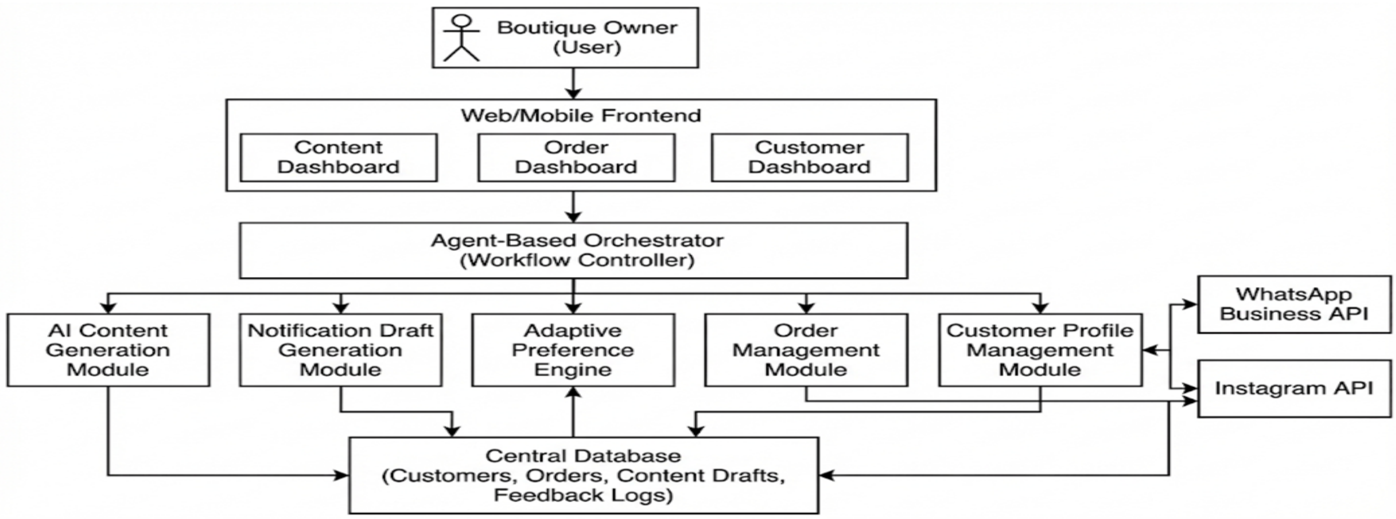

Fig 1. System Architecture

CoutureConnect is built on a multi-layered architecture integrating a React-based frontend, an agent-based orchestrator, five backend modules, a central shared database, and mocked external APIs for WhatsApp andInstagram.

Fig. 1 shows the complete system structure.

The boutique owner interacts through three dashboard views, Content, Order, and Customer, served through the frontend layer. All requests are routed through an agent-based orchestrator that acts as the workflow controller, coordinating module interactions based on predefined trigger conditions [5][6]. Five modules handle distinct responsibilities: AI ContentGeneration, Notification Draft Generation, Adaptive Preference Engine, Order Management, and Customer Profile Management. All five modules read from and write to a single central database storing customers, orders, content drafts, and feedback logs. The WhatsApp Business API and Instagram API receive dispatches from the notification and content modules respectively, and are mocked within the prototype to preserve full workflow behavior without live credentials.

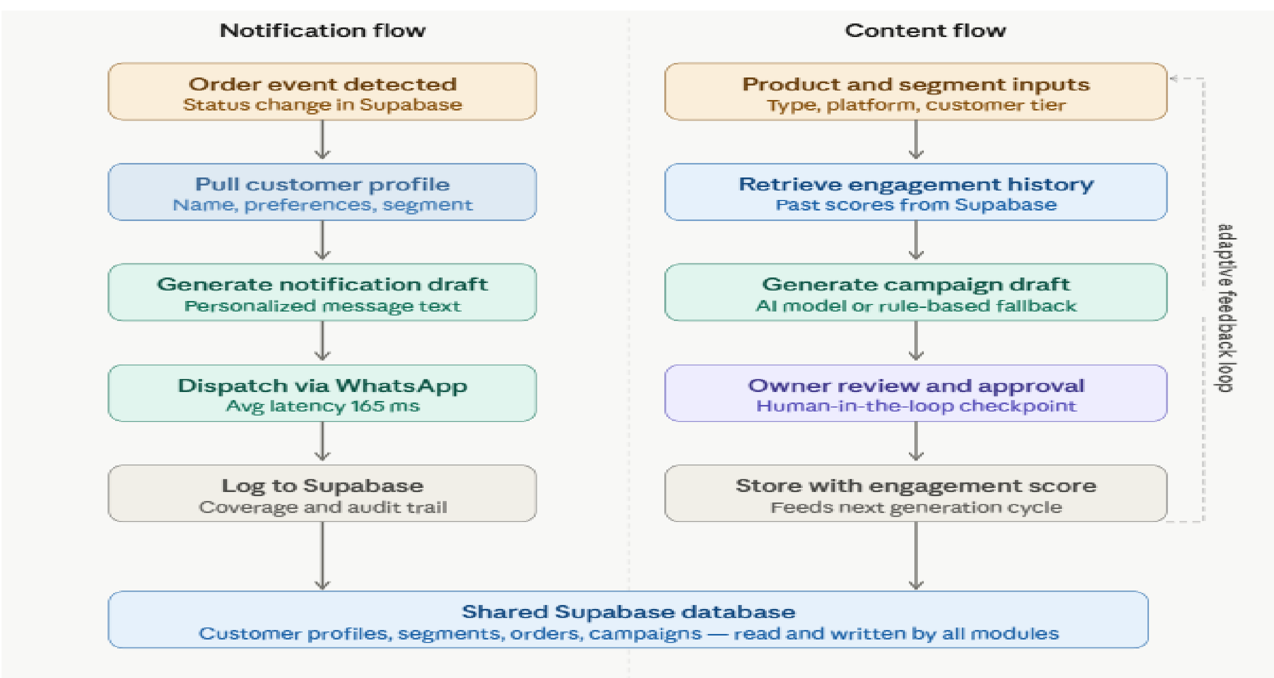

Fig .2 .Agentic Notification Flow and Adaptive Content Feedback Loop

Fig. 2 illustrates the two core operational flows. The Notification Flow runs without owner input: an order status change triggers the orchestrator, which pulls the relevant customer profile, routes the request to the Notification Draft Generation module, generates a personalized message, and dispatches it at an average latency of 165 ms.

The Content Flow begins when the owner submits a product type and target segment through the dashboard, after which the orchestrator routes the request to the AI Content Generation and Adaptive Preference Engine modules, which retrieve prior engagement scores, construct an informed prompt, and generate a draft queued for owner approval before dispatch.

The output is stored with engagement metadata so the next generation cycle can learn from it, and this adaptive feedback loop produced a 60% engagement uplift across five observed campaign cycles [9][10]. The key contribution of this architecture is that both flows are driven by the same data layer, so every order event, customer interaction, and campaign outcome compounds into richer profiles that all five modules benefit from simultaneously, producing personalization that isolated tools cannot replicate.

V. SYSTEM IMPLEMENTATION AND EXPERIMENTAL SETUP

The system was built on a decoupled architecture designed for modularity. The React frontend gives boutique owners a clean, role-aware interface for reviewing customer data, approving campaign drafts, and monitoring order activity. The backend runs across four FastAPI services, one per module, each handling its own logic while sharing a single Supabase instance. WhatsApp and Instagram APIs are mocked to preserve the full notification and publishing workflow without live credentials.

The data within Supabase is organized across three functional layers: customer profiles containing sizes, style preferences, and RFM segment labels; order and transaction records used to compute CLV and trigger notifications; and campaign history storing engagement scores retrieved by the content engine at each new generation cycle. Keeping these in one shared project means any update from one module is immediately available to all others.

The study was conducted with one boutique across five weeks of operation, covering 58 customers and 147 recorded orders. Customers were distributed across three RFM segments: 14 VIP (24%), 31 Regular (53%), and 13 At-Risk (22%), consistent with the segment distribution a small boutique with a loyal core and occasional-buyer base would naturally produce. A total of 140 trigger events were recorded across all order lifecycle and product alert scenarios.

The system was evaluated across four dimensions: weekly time savings relative to the owner's pre-deployment effort, engagement uplift across five adaptive content iterations using the formula in Section 3.10, agentic notification coverage and latency across all 140 trigger events using the formula in Section 3.8, and campaign generation reliability measured as success rate across all test runs.

VI. RESULTS AND DISCUSSION

A. *Time Savings*

Table 1 summarizes observed effort per task category before and after system deployment, computed using the formula in Section 3.9. Prior to deployment, the boutique owner reported spending eight to ten hours weekly on content drafting, order follow-ups, and record maintenance, based on self-reported time logs collected during the pre-deployment period.

Task	Manual (hrs/week)	Automated	Reduction
Content drafting	3 to 4 hours	2 minutes	85%
Order notifications	2 to 3 hours	Fully automated	100%
Customer record updates	1 to 2 hours	Auto on order event	90%
Campaign review	1 hour	Dashboard view	70%
Total	8 to 10 hours	1.2 hours	~85%

Table 1. Manual vs Automated Weekly Effort

All 140 trigger events were processed without owner input. Campaign drafts were ready in under thirty seconds across all test runs, with a generation success rate of 96%.

B. *Engagement Uplift*

The adaptive content engine was observed across five campaign cycles. Starting from generic template content, the engine incorporated engagement scores after each cycle using the formula in Section 3.7. Uplift values were computed per Section 3.10.
By the third cycle, the engine had shifted toward shorter captions and image-led formats without manual adjustment. Engagement improved from a baseline of 12.4% in cycle one to 19.8% by cycle five, representing an uplift of approximately 60%. Improvement was gradual and consistent, with no single cycle showing a dramatic jump, which reflects a realistic learning curve for small-audience engagement data rather than an ideal controlled experiment.

C. *Notification Coverage and System Performance*

Using the formula from Section 3.8, overall notification coverage across 140 trigger events was 97.9%, with 137 messages successfully dispatched. Order lifecycle events achieved full coverage. VIP product alerts reached 93% coverage and at-risk re-engagement reached 82%, with the shortfall traced to edge cases in segment-matching logic that are flagged for improvement. Average dispatch latency across all trigger types was 165 ms, keeping the system well within real-time response expectations [7][8].

D. *Customer Segmentation*

Table 2 shows the RFM-derived segment distribution across the 58-customer study cohort, with CLV values computed per Section 3.6.

Segment	Count	Share	Avg CLV
VIP	14	24.1%	Rs. 8,420
Regular	31	53.4%	Rs. 3,180
At-Risk	13	22.4%	Rs. 940
Total	58	100%	Rs. 3,760

Table 2. RFM Segment Distribution

The distribution reflects a typical small boutique: a compact high-value core, a moderate regular base, and a meaningful at-risk group that benefits most from the automated re-engagement flow. The CLV gap between VIP and At-Risk customers also validated the decision to weight the monetary dimension more heavily in the RFM formula.

E. *Discussion*

The results confirm that CoutureConnect addresses each problem it was designed for. Content creation is automated and improves over successive iterations. Customer communication runs without manual input across all order and product events. Customer data is centralized and actively drives both notification personalization and content targeting simultaneously, which neither component could achieve independently [8].
The human-in-the-loop approval panel kept the owner in control of outward-facing communication throughout the study, which was important for building trust in the automation early on. The owner reported that reviewing and approving a draft took under two minutes per campaign, compared to thirty or more minutes for drafting one from scratch.
The 60% engagement uplift is a meaningful improvement for a boutique audience of this size, but it should be interpreted carefully.

With a small follower base, individual post variance is high, and a larger observation window would be needed to confirm the trend with confidence. Similarly, the simplified CLV formula works well at this scale but may need refinement for boutiques with more complex buying patterns [12]. Rule-based agent logic also limits the system to trigger scenarios defined at design time, which is a known constraint for future development.

VII. CONCLUSION

This paper presents CoutureConnect, a modular AI-based system designed to support small boutique retailers by automating key operational tasks. The system combines an Adaptive Content Engine with an Agentic Customer Flow, both connected through a shared customer data layer, allowing content generation, communication, and customer management to work together seamlessly.

The evaluation showed a clear improvement in efficiency and engagement. The system reduced weekly manual effort by around 80–85%, improved content engagement across iterations, and achieved high automation coverage (~98%) for notification workflows. While these results are based on simulated data, they demonstrate the practical potential of the system in real-world boutique environments.

More importantly, the system highlights that boutique businesses do not require complex or enterprise-level tools. Instead, they benefit from simple, integrated solutions that match their daily workflows, reduce repetitive tasks, and improve consistency in customer interaction.

CoutureConnect shows that such a system is both feasible and effective, even at a small operational scale. Future work will focus on real-world deployment and further improving adaptability and personalization through advanced learning mechanisms.

VIII. ACKNOWLEDGMENT

The authors would like to thank their project mentor and coordinator for their guidance and feedback throughout this work. They are also grateful to the Academic Director and the Department of Computer Science and Engineering at Keshav Memorial Institute of Technology for providing the resources and environment that made this project possible.

REFERENCES

[1] Grewal, D., Roggeveen, A. L., and Nordfalt, J. (2020). The future of retailing. *Journal of Retailing*, 96(1), 1-11.
[2] Verhoef, P. C., et al. (2021). Digital transformation: A multidisciplinary reflection and research agenda. *Journal of Business Research*, 122, 889-901.
[3] Huang, M. H., and Rust, R. T. (2018). Artificial intelligence in service. *Journal of Service Research*, 21(2), 155-172.
[4] Kumar, V., et al. (2019). Understanding the role of artificial intelligence in personalized engagement marketing. *California Management Review*, 61(4), 135-155.
[5] Russell, S., and Norvig, P. (2020). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
[6] Wooldridge, M., and Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2), 115-152.
[7] Xu, A., et al. (2017). A new chatbot for customer service on social media. *Proceedings of the ACM CHI Conference*, 1-10.
[8] Yao, S., et al. (2023). ReAct: Synergizing reasoning and acting in language models. *Proceedings of ICLR 2023*.
[9] Li, L., et al. (2010). A contextual-bandit approach to personalized news article recommendation. *Proceedings of WWW 2010*, 661-670.
[10] Kim, A. J., and Ko, E. (2012). Do social media marketing activities enhance customer equity? *Journal of Business Research*, 65(10), 1480-1486.
[11] Payne, A., and Frow, P. (2005). A strategic framework for customer relationship management. *Journal of Marketing*, 69(4), 167-176.
[12] Reinartz, W., Krafft, M., and Hoyer, W. D. (2004). The customer relationship management process. *Journal of Marketing Research*, 41(3), 293-305.
[13] Bull, C. (2003). Strategic issues in customer relationship management implementation. *Business Process Management Journal*, 9(5), 592-602.
[14] Appel, G., et al. (2020). The future of social media in marketing. *Journal of the Academy of Marketing Science*, 48(1), 79-95.
[15] Hevner, A. R., et al. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75-105.