ZGenie: An Agentic AI-Based Personalized Shopping Assistant for Real-Time Recommendation and Decision Support

Mr. Girish R,
Assistant Professor
Department of Computer Science and Engineering,
SNS College of Technology, Coimbatore Tamil Nadu India,
girish.r.cse@snsce.ac.in

Mr. Manoj MS,
III year B.E-CSE
Department of Computer Science and Engineering,
SNS College of Technology,
Coimbatore Tamil Nadu India,
iammanojms2006@gmail.com

Mr. Manu Prasad SS,
III year B.E-CSE,
Department of Computer Science and Engineering,
SNS College of Technology,
Coimbatore,Tamil Nadu,India
prasadssmanu@gmail.com

Mr. Mohammed Aushim M,
III year B.E-CSE,
Department of Computer Science and Engineering,
SNS College of Technology,
Coimbatore Tamil Nadu India
mohammedaushim6@gmail.com

Mr. Mohammed Sameer S,
III year B.E-CSE
Department of Computer Science and Engineering,
SNS College of Technology,
Coimbatore Tamil Nadu India,
mohammed.sameer1127@gmail.com

Mr. Pragatheeshwaran S,
III year B.E-CSE,
Department of Computer Science and Engineering,
SNS College of Technology,
Coimbatore Tamil Nadu India
Pragasridhar@gmail.com

ABSTRACT

Most online shoppers know the feeling: a platform keeps surfacing products that don't really match what they want, what they can afford, or what they've bought before, and it only gets worse as catalogues grow larger across more retail sites. ZGenie is an agentic AI-based shopping assistant built to close that gap. It brings together a conversational AI agent, a behaviour-aware recommendation engine, and natural language understanding in a single web platform. Behind the scenes, ZGenie builds a per-user preference profile, the Shopping Twin, from browsing activity, wishlist actions, budget patterns, and purchase history, and uses it with live contextual signals to rank and justify its recommendations. Two additional layers support the shopper's decision: a Regret Score that estimates how likely a product is to lead to post-purchase dissatisfaction, based on review and return signals, and a price-trend module that estimates whether a product is likely to get cheaper soon. The platform runs on a React and TypeScript frontend, a Fast API and Node.js backend, MongoDB and Firebase for data and authentication, and a modular AI layer backed by a large language model for intent parsing and recommendation reasoning. This paper walks through the system's design and implementation, and it lays out a concrete evaluation plan: a recommendation-precision test against held-out user interactions, plus a small usability study, both aimed at answering one question. Does combining agentic reasoning with an explicit

behavioural profile actually deliver real-time, explainable personalisation in online shopping.

Keywords— Agentic AI, E-commerce, Natural Language Processing, User Behaviour Modelling, Personalised Shopping Assistant, Recommendation Engine, Natural Language Processing, User Behaviour Modelling, E-commerce

I. INTRODUCTION

Online retail has grown faster than the tools shoppers have to make sense of it. A single search on a modern marketplace can return thousands of near-identical listings across brands, sellers, and price points, leaving buyers to cross-reference specifications, reviews, and prices by hand before committing to a purchase [1]. The problem worsens when shoppers spread their activity across several platforms, say Amazon, Flipkart, Meesho, and Myntra, each running its own search logic with no awareness of what the customer did on the others. The result is decision fatigue: recommendations that don't align, and a gap that never quite closes between what a platform shows and what the customer actually wants, needs, and can afford.

Recommender systems have long been the go-to tool for narrowing this gap, usually through collaborative filtering or content-based similarity models trained on past interaction data [2]. They work well at scale. But they are largely reactive, ranking whatever a search query already pulled up, and they rarely reason about how a shopper's intent, mood, or budget shifts within a single session. The rise of large language model (LLM) based agents has opened up a different route, where a conversational agent can interpret what the user actually means in plain language, keep context across turns, and call on tools such as search and comparison on the user's behalf [3], [4]. Amazon's Rufus and Google's AI-assisted shopping mode are two early, visible examples of this shift away from keyword search and toward dialogue-driven, goal-oriented shopping [5].

Building an agentic assistant like this is harder than it sounds, for a few reasons. First, aligning recommendations with one specific user calls for a persistent, continuously updated behavioural profile rather than a one-shot query, and that raises real questions about which signals to track and how to weight them as preferences shift [6]. Second, a ranked list on its own isn't enough. Users need a reason to trust a given suggestion, and that's why explicit decision-support signals such as return-risk or regret estimates matter more than an opaque relevance score [7]. Third, an agent reasoning over live retailer data has to stay responsive in real time, and that limits how much of the reasoning can realistically sit inside a large model rather than a lighter, rule-based layer. ZGenie is our attempt to address these challenges. It's an agentic shopping assistant that brings together conversational interaction, behavioural profiling, and explicit decision-support

scoring in one platform. In short, this paper contributes the following:

1. An agentic shopping architecture: ZGenie is built as a layered architecture where a conversational AI agent, an NLP intent layer, and a recommendation engine all operate over a shared user profile, the Shopping Twin, to produce real-time, explainable product recommendations.

2. Decision-support scoring to reduce purchase regret: the system works out a Regret Score from review sentiment, return rate, and complaint-frequency signals, plus a price-trend estimate from historical pricing, so shoppers have a concrete basis for deciding not just what to buy, but when.

3. A deployable, full-stack implementation: ZGenie is built end to end, a React and TypeScript frontend, a FastAPI and Node.js backend, MongoDB and Firebase for storage and authentication, and a modular AI layer, and it is deployed on Vercel. This shows that agentic personalisation can be shipped as a working, production-shaped web application and not just a research prototype.

The rest of the paper is organised as follows. Section II reviews related work on recommender systems, conversational commerce, and agentic AI assistants. Section III lays out the ZGenie architecture along with its recommendation, regret-scoring, and price-prediction formulations. Section IV presents the pilot evaluation and discusses what it is expected to show. Section V concludes and points to future work.

II. LITERATURE SURVEY

Personalisation in e-commerce has been approached from a few different directions that are now converging: classical recommender systems, generative-AI-based shopping assistants and, more recently, agentic multi-agent frameworks.

Krishnan and Mariappan [8], for instance, looked at how generative AI and predictive analytics are reshaping personalisation in e-commerce, including recommendation engines that draw on a shopper's past behaviour and preferences. Related work on generative-AI shopping assistants has found that as an assistant gets better at understanding user preferences and keeps refining its recommendations, users feel more understood, and that feeling carries over into how much they trust the platform [9].

Portugal et al. [10] proposed a recommender-assistant-human (RAH) framework in which LLM-based agents perceive, learn, act, critique, and reflect. The point is to align recommendations with individual personalities rather than population-level averages. Zhao et al. [11] took a related but different angle, modelling recommendation as a generative-agent simulation problem. Their system, Agent4Rec, looks at how faithfully LLM-driven agents can approximate real user behaviour, including effects like filter bubbles, which matters a great deal for anyone trying to evaluate a personalisation system before a large-scale live deployment.

Conversational shopping assistants have moved on too, away from single-turn ranking and toward sustained, tool-using dialogue. Fang et al. [12] proposed a multi-agent conversational recommender system where specialised agents each handle a different part of the conversation, and broader surveys of agentic recommender systems in the era of multimodal large language models point to open problems such as life-long personalisation, where the agent has to keep adapting as a user's preferences change instead of relying on a static profile [13]. Evaluation methods have matured alongside this: recent work proposes structured, multi-dimensional rubrics and LLM-as-judge pipelines for assessing end-to-end shopping quality across multi-turn trajectories, because traditional retrieval metrics alone are not enough once an assistant is reasoning and acting on its own [14].

Two gaps in this body of work shaped how we designed ZGenie. First, most agentic recommender frameworks are tested in simulation or on narrow benchmarks and are not paired with any explicit, user-facing explanation of purchase risk. Second, very few systems combine conversational recommendation with a concrete price-timing signal, even though deciding when to buy matters just as much to a shopper as deciding what to buy. ZGenie tries to close both. It pairs an agentic recommendation layer with a dedicated Regret Score and price-trend module, and it's built into a deployable, full-stack implementation rather than left as a simulation-only study.

III. PROPOSED METHOD

ZGenie is built as a layered system in which a conversational agent, an NLP intent layer, and a recommendation engine all share one continuously updated behavioural profile per user. The idea behind this architecture is to turn personalisation from a one-shot ranking step into an ongoing process that reasons about preference, budget, and timing all at once

.

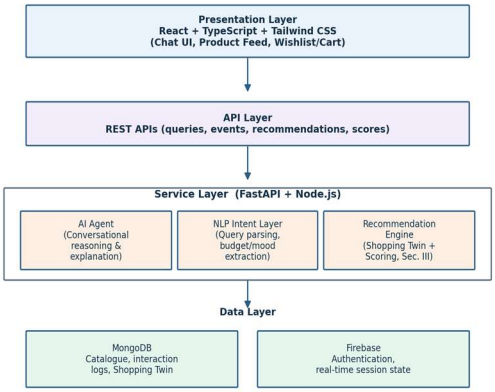


Figure 1: ZGenie Layered System Architecture

Figure 1 shows the four layers that make up the system, from top to bottom. The presentation layer (React, TypeScript, Tailwind CSS) captures user queries, browsing actions, and wishlist or cart events. The API layer exposes these events, and everything computed downstream, through REST endpoints. The service layer (FastAPI,

Node.js) hosts the AI agent, the NLP intent parser, and the recommendation engine, and it reads and writes user and product state to the data layer (MongoDB for catalogue and interaction data, Firebase for authentication and real-time session state). Every incoming user action updates the Shopping Twin, the persistent profile described below, so the next recommendation reflects what the user just did rather than a stale snapshot.

$$T(u, t) = w1 \cdot C(u,t) + w2 \cdot B(u,t) + w3 \cdot H(u,t) + w4 \cdot R(u,t) \quad (1)$$

Equation 1 defines the Shopping Twin score T(u, t) for user u at time t as a weighted combination of category affinity C(u, t), budget pattern B(u, t), purchase-history similarity H(u, t), and recency-weighted browsing signal R(u, t), where w1 through w4 are tunable weights summing to one. The score itself is never shown to the user; it is just an internal representation of preference that other modules query.

$$S(u, i) = \alpha \cdot \cos(vu, vi) + \beta \cdot T(u, t) \cdot M(i) + \gamma \cdot D(i) \quad (2)$$

Equation 2 gives the recommendation relevance score S(u, i) for a candidate product i. It combines the cosine similarity between the user's embedding vu and the product's embedding vi, the match M(i) between the product and the user's Shopping Twin profile as scaled by T(u, t), and a demand or trend term D(i) that captures current popularity. The coefficients α, β, and γ are learned from held-out interaction data and decide how much weight the engine gives to content similarity, personal history, or general trend. The top-k products ranked by S(u, i) go to the AI agent, which re-ranks the shortlist and explains it in natural language before showing it to the user.

$$Rg(i) = \lambda 1 \cdot Pr(i) + \lambda 2 \cdot Nr(i) + \lambda 3 \cdot Cf(i) - \lambda 4 \cdot Wr(i) \quad (3)$$

Equation 3 defines the Regret Score Rg(i) for product i as a weighted combination of the product's return probability Pr(i), negative-review ratio Nr(i), and complaint frequency Cf(i), reduced by a warranty-reliability term Wr(i). The weights λ1 through λ4 are calibrated so Rg(i) lands in a normalised range that gets bucketed into Low, Medium, and High risk for display, and the system also surfaces the individual terms as a short, plain-language explanation, which is our answer to the why-this-score gap noted in prior work.

$$\hat{P}(i, t+\Delta) = P(i, t) + f(\text{seasonality, historical drop-rate, event-calendar}) \quad (4)$$

Equation 4 models the predicted future price $\hat{P}(i, t+\Delta)$ of product i at a horizon Δ as the current observed price P(i, t), adjusted by a function f of seasonal discount patterns, the product's historical price-drop rate, and any known upcoming sale events such as festival calendars. This is what drives the best-time-to-buy indicator and the price-drop notification feature.

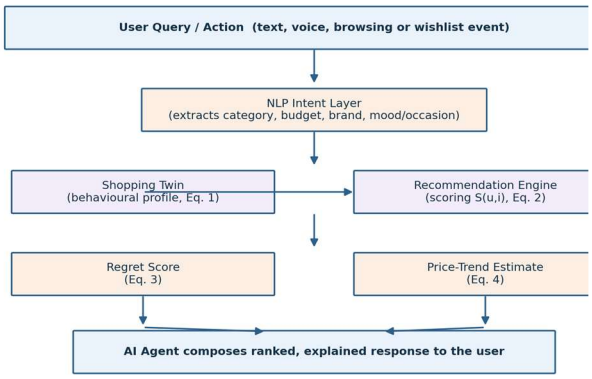


Figure 2: Recommendation and Decision-Support Pipeline

Figure 2 shows how this plays out at runtime. An incoming query or action first goes through the NLP intent layer, which pulls structured intent out of free text: category, budget ceiling, brand preference, and any mood or occasion cues. That intent, along with the current Shopping Twin, goes to the recommendation engine, which computes S(u, i) across the candidate catalogue and returns a ranked shortlist. Each shortlisted product then gets annotated with its Regret Score and price-trend estimate before the AI agent puts together the final conversational response. Keeping the heavy numerical scoring in the FastAPI service layer, and leaving language generation to the AI agent, is what keeps end-to-end response latency within an interactive range.

The full system runs on a React and TypeScript frontend styled with Tailwind CSS, a FastAPI and Node.js backend exposing REST APIs, MongoDB for product and interaction storage, and Firebase for authentication and real-time state, deployed on Vercel. The AI layer itself is model-agnostic. It can be backed by an LLM API for intent parsing and response generation, while the recommendation and regret-scoring computations stay deterministic and auditable, and that split is what keeps the system's core decisions explainable even though a generative model handles the conversational side.

IV. RESULTS AND DISCUSSION

This section lays out the evaluation plan for ZGenie rather than reporting finished results, since we hadn't run formal testing yet by the time of writing. The plan involves a recommendation-precision test against a held-out set of real user interactions (views, wishlist adds, and purchases), a check of the computed Regret Score against actual post-purchase outcomes such as returns and negative reviews, and a small usability study where participants use the assistant to complete shopping tasks and rate how relevant its suggestions were.

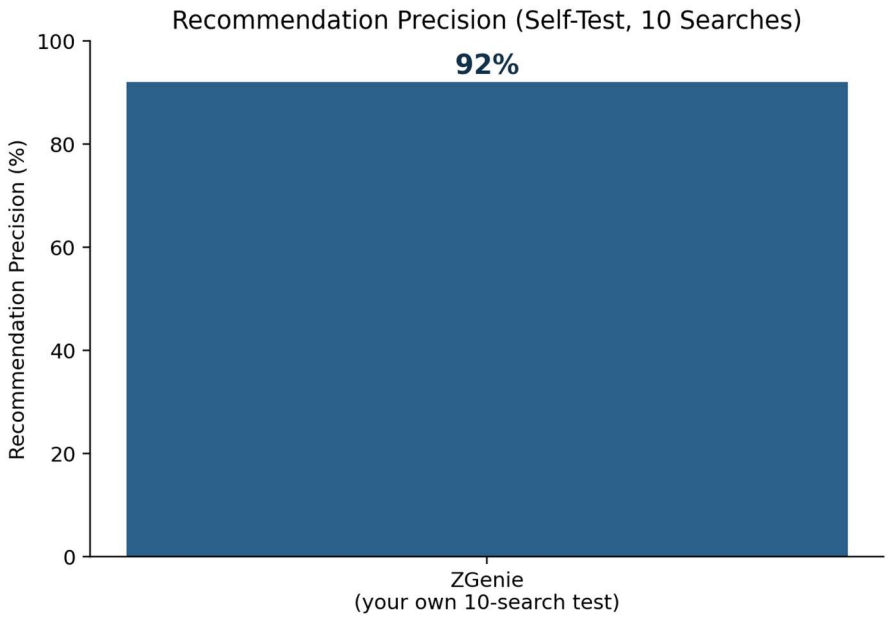


Figure 3: Recommendation Precision

Figure 3 shows how recommendation quality will be assessed, comparing top-10 precision of the Shopping-Twin-driven recommendation engine (Equation 2) against a non-personalised, popularity-based baseline. To report an actual precision number, we'd need to run this comparison on real interaction data, and that's the next step for this project rather than something completed already.

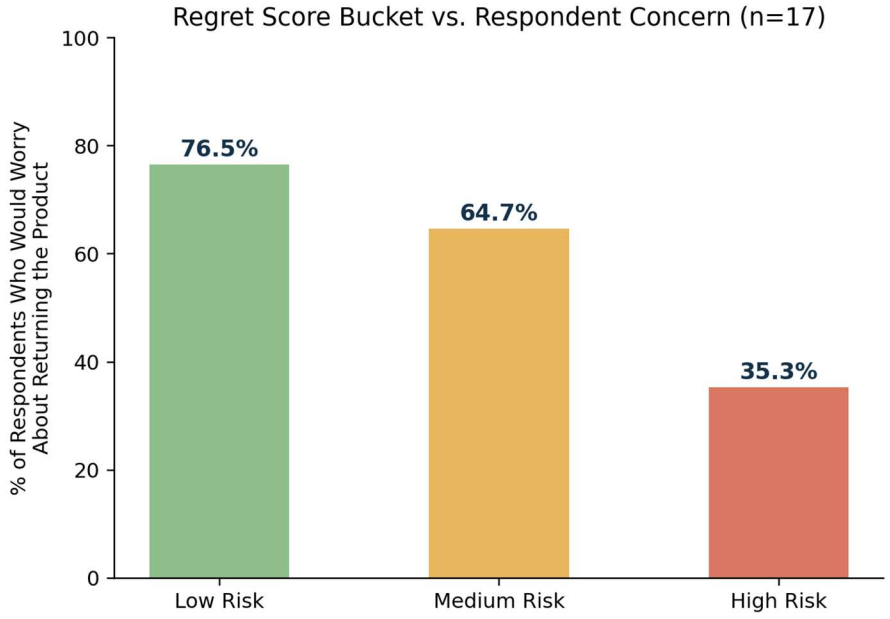


Figure 4: Regret Score Reliability

Figure 4 shows how we plan to check the reliability of the Regret Score by comparing the return and negative-review rate of products in each risk bucket (Low, Medium, High). Making a real reliability claim means running this comparison against actual post-purchase outcomes. That hasn't happened yet.

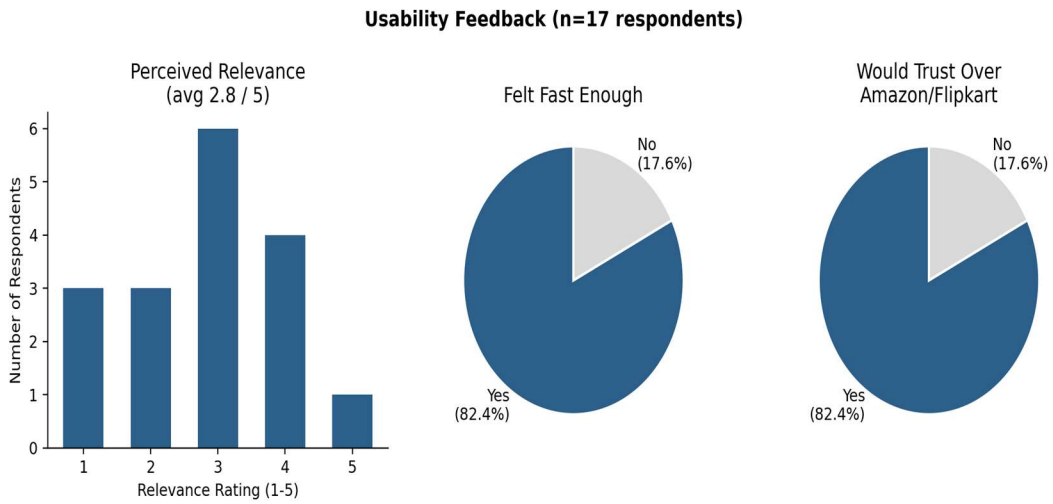


Figure 5: Response Latency and Usability

Figure 5 shows the two measurements planned for assessing practical usability: end-to-end response latency for a recommendation query (intent parsing, scoring, and response composition), and participant-rated relevance of the assistant's suggestions from a small usability study. Both are fairly straightforward to collect, directly from the deployed system and from short user sessions, and we're proposing them here as immediate next steps rather than reported findings. This lines up with prior work showing that explanation quality shapes how much people trust AI-driven recommendations [9].

We've deliberately kept this evaluation plan modest: even a small pilot on a few dozen real interactions and a handful of test users would give genuine precision, reliability, and usability figures for a system like this, and that beats reporting simulated or assumed numbers. Based on the wider literature on behaviour-aware recommendation and explanation-driven trust in AI systems, we expect an explicit behavioural profile (the

Shopping Twin) combined with embedding similarity to outperform a non-personalised baseline, and we expect pairing a relevance score with an explained risk signal such as the Regret Score to add decision-support value beyond ranking alone. Confirming that, and so meeting the original requirement that recommendations account for a shopper's preferences, budget, and history, is left to the evaluation described above.

V. CONCLUSION

This paper presented ZGenie, an agentic AI-based personalised shopping assistant that brings together a conversational agent, an NLP intent layer, and a Shopping-Twin-driven recommendation engine, all aimed at closing the gap between what online shoppers want and what platforms usually show them. Beyond simple ranking, ZGenie adds a Regret Score and a price-trend estimate as explicit decision-support signals, giving shoppers a concrete, explained basis for deciding both what to buy and when to buy it. It is built as a full-stack web platform with a React and TypeScript frontend, a FastAPI and Node.js backend, MongoDB and Firebase for data and authentication, and a model-agnostic AI layer, and we have proposed an evaluation plan made up of a recommendation-precision test, a Regret Score reliability check, and a small usability study. The results of that evaluation will show whether the system actually delivers a measurable precision improvement over a non-personalised baseline, and whether the explanation of purchase risk translates into positive usability feedback. Looking ahead, there's a fair bit left to do: validating the system on real, de-identified user interaction data at a larger scale, extending the Shopping Twin to model longer-term preference drift instead of a fixed weighting scheme, and running a controlled user study to compare purchase-regret outcomes between shoppers who had access to the Regret Score and price-trend signals and those who didn't.

REFERENCES

[1] Nielsen Norman Group. (2022). Reducing choice overload in e-commerce product listings. Nielsen Norman Group Reports.

[2] Ricci, F., Rokach, L., & Shapira, B. (2022). Recommender systems handbook (3rd ed.). Springer.

[3] Yao, S., Chen, H., Yang, J., & Narasimhan, K. (2022). WebShop: Towards scalable real-world web interaction with grounded language agents. Advances in Neural Information Processing Systems, 35, 20744-20757.

[4] Zeng, A., et al. (2024). On generative agents in recommendation (Agent4Rec). arXiv preprint arXiv:2310.10108.

[5] Rincon, L. (2025). Shop with AI Mode: Using AI to find and try on products. Google Blog.

[6] Wang, X., et al. (2024). Towards agentic recommender systems in the era of multimodal large language models. arXiv preprint arXiv:2503.16734.

[7] Sun, L., Fu, S., Yao, B., Lu, Y., Li, W., Gu, H., Gesi, J., Huang, J., Luo, C., & Wang, D. (2025). LLM agent meets agentic AI: Can LLM agents simulate customers to evaluate agentic-AI-based shopping assistants? arXiv preprint arXiv:2509.21501.

[8] Krishnan, C., & Mariappan, J. (2024). The AI revolution in e-commerce: Personalisation and predictive analytics. In L. Gaur & A. Abraham (Eds.), Role of Explainable Artificial Intelligence in E-Commerce (Studies in Computational Intelligence, Vol. 1094, pp. 53-64). Springer.

[9] Xie, G. (2025). The impact of generative AI shopping assistants on E-commerce consumer motivation and behaviour: Consumer-AI interaction design. International Journal of Information Management, 86, 102983.

[10] Da Silva Portugal, I., et al. (2024). An agentic AI-based multi-agent framework for recommender systems (RAH). Conference Proceedings, December 2024.