IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

Smart Deal Finder BoT Using Agentic AI: An Intelligent Conversational System for Automated E-Commerce Price Discovery and Comparison

Navnath Lahane[1], Aditya Patil[2], Madhav More[3] Prof. Anil Walke[4],
Department of Artificial Intelligence and Data Science[1,2,3],
Project Guide: Department of Artificial Intelligence and Data Science[4],
ISBM College of Engineering, Pune, Maharashtra, India
lahanenavnath14@gmail.com[1], adityapatil2843@gmail.com[2], madhavmore23445@gmail.com[3]
anil.walke@isbmcoe.org[4]

Abstract: *The rapid expansion of e-commerce has given consumers access to millions of products across platforms such as Amazon, Flipkart, and eBay, but identifying the best available deal among these platforms remains a slow and largely manual task. Prices, discounts, delivery charges, and ratings vary continuously across sellers, and existing comparison tools generally return static, non-personalized results without conversational support. This paper presents the Smart Deal Finder Bot, an autonomous shopping assistant built on Agentic AI principles that understands natural-language product queries, retrieves live product data from multiple e-commerce sources through APIs and web scraping, and autonomously reasons over price, discount, rating, and delivery parameters to recommend the most suitable deal.*

The system architecture integrates a Natural Language Processing (NLP) module for intent and entity extraction, a web-scraping/API integration layer for real-time data acquisition, a multi-parameter comparison engine, an Agentic AI decision layer for goal-oriented recommendation, and a database layer for persistence of user history and analytics. The prototype was implemented using Python, FastAPI/Flask, React.js/Streamlit, BeautifulSoup, Selenium, spaCy, and MongoDB/MySQL, and deployed on cloud infrastructure. Experimental evaluation across multiple product categories shows an intent-recognition accuracy of 94%, an overall recommendation accuracy of approximately 92%, an average response time of 2 to 4 seconds, and a scraping success rate of about 90%, confirming that the system meaningfully reduces manual comparison effort while improving transparency in purchasing decisions. The paper concludes by identifying current limitations, primarily around API rate limits and the fragility of web scraping against anti-bot defenses, and outlines future work on voice interaction, multilingual support, predictive pricing, and broader platform coverage.

Keywords: Agentic AI; Natural Language Processing; E-Commerce Price Comparison; Conversational Chatbot; Web Scraping; Recommendation System.

I. INTRODUCTION

Online shopping has become one of the dominant modes of retail commerce worldwide, with platforms such as Amazon, Flipkart, eBay, and Myntra offering millions of products from competing sellers. While this growth has expanded consumer choice, it has simultaneously made the task of finding the best deal considerably harder: identical or near-identical products are frequently listed at different prices, with different discounts, delivery charges, and seller ratings across platforms. Manually checking each marketplace before a purchase decision is time-consuming, error-prone, and increasingly impractical given how quickly prices and offers change.

DOI: 10.48175/IJARSCT-37206

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

Conventional price-comparison websites attempt to address this problem, but they typically present static, list-based results scraped or pulled at infrequent intervals. They generally lack conversational interaction, cannot interpret free-form natural-language requests, and do not adapt recommendations to an individual user's stated priorities (for example, prioritising delivery speed over price, or rating over discount). Advances in Artificial Intelligence (AI), Natural Language Processing (NLP), and the emerging paradigm of Agentic AI — where autonomous agents reason, plan, and act toward a goal with minimal human supervision — create an opportunity to close this gap.

This paper proposes and evaluates the Smart Deal Finder Bot, a conversational shopping assistant that combines NLP-based query understanding with an Agentic AI decision layer to autonomously search, compare, and recommend products across multiple e-commerce platforms in real time. The specific objectives of this work are to: (i) design a chatbot capable of interpreting natural-language shopping queries; (ii) build a data-acquisition layer that gathers live product information via APIs and web scraping; (iii) implement a multi-parameter comparison and ranking engine; (iv) implement an Agentic AI module that autonomously selects the best deal according to user-specified or inferred priorities; and (v) evaluate the resulting system for accuracy, responsiveness, and reliability.

The remainder of this paper is organized as follows. Section 2 reviews related work in AI-powered chatbots, price-comparison systems, conversational recommendation, and Agentic AI. Section 3 describes the proposed methodology and system architecture. Section 4 presents implementation details. Section 5 reports experimental results and discussion. Section 6 concludes the paper and outlines future work.

II. LITERATURE REVIEW

Research relevant to this work spans four overlapping areas: AI-powered conversational chatbots in retail, intelligent price-comparison systems, conversational search and recommendation, and the recent emergence of Agentic AI as a design paradigm for autonomous digital assistants.

2.1 AI-Powered Chatbots in E-Commerce

Sharma et al. examined how conversational AI systems built around NLP and machine learning improve customer engagement and reduce response latency in online retail, but their model was oriented toward customer support rather than cross-platform price comparison. Patel and Rao compared chatbot frameworks including Rasa, Dialogflow, and BERT-based conversational models, demonstrating that modern NLP pipelines can generate context-aware responses, though their work was not designed specifically for e-commerce deal discovery. These studies collectively establish that conversational AI improves user engagement, but also reveal a gap: none of the reviewed systems autonomously analyse multi-platform pricing data to recommend a specific deal.

2.2 Intelligent Price Comparison Systems

Li and Kumar developed a system that uses web scraping to extract prices, discounts, and specifications from multiple e-commerce sites, highlighting the importance of data normalization but lacking conversational interaction or personalization. Garcia et al. proposed a machine-learning approach using regression models on historical pricing data to predict favourable buying windows; while useful for trend forecasting, this approach does not support live conversational assistance or real-time comparison. Together, these works confirm the technical feasibility of automated price extraction but show that comparison alone, without an interactive and adaptive layer, leaves the user to interpret the results manually.

2.3 Conversational Search and Recommendation Systems

Singh et al. built a chatbot using semantic search and NLP to improve product retrieval and query understanding, improving user engagement but stopping short of multi-platform price comparison or deal analysis. Broader recommendation-engine research has likewise tended to optimize for user-preference matching using collaborative or content-based filtering while largely ignoring real-time, cross-platform pricing signals. The Smart Deal Finder Bot

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

proposed in this paper is positioned to combine these two historically separate threads — conversational recommendation and live price comparison — into a single pipeline.

2.4 Agentic AI in Intelligent Systems

Agentic AI refers to systems in which an autonomous agent can reason, plan, invoke tools, and adapt its actions to satisfy a goal with limited human intervention, as opposed to traditional chatbots that primarily map input patterns to predefined responses. Recent work on autonomous digital assistants demonstrates that agent-based architectures — typically combining a large-language-model-based reasoning core, short- and long-term memory, and a tool-invocation layer for API calls and external system access — can independently gather information, evaluate trade-offs, and execute multi-step tasks. This project applies that architecture to the shopping domain, using an agent to evaluate price, rating, delivery, and discount trade-offs and to generate an optimized, goal-directed recommendation.

2.5 Research Gap

The reviewed literature shows considerable progress in conversational chatbots, scraping-based comparison tools, and recommendation engines individually, but very few systems unify conversational AI, live multi-platform price comparison, and autonomous Agentic AI decision-making into a single integrated assistant. This paper addresses that gap by proposing a system that combines NLP-based query understanding, real-time data acquisition, a comparison engine, and an Agentic AI reasoning layer within one conversational interface.

III. METHODOLOGY / PROPOSED SYSTEM

3.1 System Overview

The Smart Deal Finder Bot is structured as a modular pipeline in which a conversational front end accepts a natural-language query, an NLP module extracts structured intent and entities, a data-acquisition layer retrieves live product listings from multiple e-commerce platforms, a comparison engine scores candidate products on several weighted parameters, and an Agentic AI layer autonomously selects and explains the recommended deal. The high-level workflow proceeds as follows:

1. The user submits a product-related query through the chatbot interface (e.g., "Find the best Samsung mobile under ₹30,000").
2. The NLP engine tokenizes the query, removes stop words, performs lemmatization, and applies Named Entity Recognition (NER) to extract the brand, product category, and budget.
3. An intent classifier determines whether the query corresponds to a search, a price comparison, or a best-deal recommendation request.
4. The web-scraping/API integration layer queries multiple e-commerce platforms in parallel for matching listings.
5. Retrieved data is cleaned, deduplicated, and normalized (currency formatting, missing-value handling).
6. The comparison engine scores each candidate product using a weighted combination of price, discount, rating, delivery speed, and seller reputation.
7. The Agentic AI decision layer evaluates the ranked candidates against the user's explicit or inferred priorities and autonomously selects the recommended deal.
8. The chatbot returns a conversational response containing the recommended product, its price, savings, and a direct purchase link, and logs the interaction to the database for history and analytics.

3.2 NLP Query Understanding

Given an input such as "Find best Samsung laptop under ₹50,000", the NLP module performs tokenization to split the sentence into discrete tokens, removes non-informative stop words such as "find" and "under", and applies NER to recognise the brand ("Samsung"), the product category ("laptop"), and the budget ceiling ("₹50,000"). An intent

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

classifier then labels the query — in this example, as a price-comparison intent. Table 1 illustrates a representative extraction.

Parameter	Extracted Value
Brand	Samsung
Product	Laptop
Budget	₹50,000
Intent	Price Comparison

Table 1. Example output of the NLP extraction pipeline.

3.3 Data Acquisition Layer

Once structured query parameters are available, the system retrieves live listings from supported marketplaces — Amazon, Flipkart, eBay, Snapdeal, and Myntra — using a combination of REST APIs (where available) and web scraping via BeautifulSoup, Scrapy, and Selenium. For each platform the system extracts the product title, price, discount percentage, rating, number of reviews, delivery charge, estimated delivery time, seller name, and product URL. Because identical fields differ in format across sites, a normalization step converts all values to consistent units (e.g., a single currency representation and a 0–5 rating scale) before comparison.

3.4 Product Comparison Engine

The comparison engine assigns a weighted score to each candidate listing using price, discount, rating, delivery charge, and seller reputation as inputs, with price carrying the largest weight by default. Table 2 shows a representative comparison across three platforms for a single query.

Platform	Price (₹)	Discount	Rating
Amazon	54,999	10%	4.5
Flipkart	53,499	12%	4.4
eBay	55,200	8%	4.3

Table 2. Representative multi-platform product comparison output.

Ranking proceeds in five stages: data collection, normalization, duplicate removal, weighted-feature scoring, and final ranking. The output of this stage is an ordered list of candidate products together with the explicit trade-offs (e.g., lowest price versus highest rating) between the top few options.

3.5 Agentic AI Decision Layer

The Agentic AI module is the component that distinguishes this system from a conventional comparison tool. Rather than simply returning a sorted list, the agent reasons over the ranked candidates in light of the user's stated or inferred priorities — for example, favouring the fastest delivery, the highest rating, or the lowest price — and autonomously commits to a single recommended deal, which it can justify in natural language. Architecturally, the agent combines a planning and reasoning core with short-term memory (the current conversation context), long-term memory (prior searches and preferences), and a tool-invocation layer that issues the API calls and scraping requests needed to gather supporting evidence. This design allows the agent to adapt its recommendation dynamically as new information becomes available, rather than relying on a fixed, pre-programmed decision rule.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

3.6 System Architecture and Modules

The complete system is organized into the following layers, each implemented as an independently deployable module communicating over internal REST APIs:

- User Interface Layer — a conversational chatbot front end built with React.js (or Streamlit for the prototype), supporting product search, comparison-table display, and search-history review.
- NLP Processing Layer — tokenization, stop-word removal, lemmatization, NER, and intent classification using spaCy and NLTK.
- Agentic AI Decision Layer — the autonomous reasoning core described in Section 3.5.
- Web Scraping / API Integration Layer — BeautifulSoup, Scrapy, Selenium, and REST API clients for live data acquisition.
- Product Comparison Engine — weighted multi-parameter scoring and ranking.
- Recommendation Engine — generation of the final personalized suggestion and supporting savings information.
- Database Layer — MongoDB/MySQL storage of user profiles, search history, product data, and recommendation logs.
- Admin Dashboard Layer — monitoring of system usage, scraping performance, and recommendation statistics.
- Security Layer — JWT-based authentication, HTTPS, and data encryption.
- Cloud Deployment Layer — containerized deployment on AWS, Heroku, or Google Cloud for scalability and availability.

IV. IMPLEMENTATION

4.1 Technology Stack

The prototype was implemented primarily in Python, using FastAPI (and Flask/Django for earlier iterations) on the backend and React.js/Streamlit on the frontend. NLP processing relies on spaCy and NLTK; data acquisition uses BeautifulSoup, Scrapy, and Selenium; data analysis uses Pandas and NumPy; and persistence is handled by MongoDB or MySQL. The application is containerized and deployed on cloud infrastructure (Railway/AWS/Heroku/Google Cloud), with Redis used for caching and Celery for background task scheduling.

Component	Technology / Tool
Backend Framework	FastAPI / Flask / Django
Frontend	React.js / Streamlit
NLP	spaCy, NLTK
Web Scraping	BeautifulSoup, Scrapy, Selenium
Data Processing	Pandas, NumPy
Database	MongoDB / MySQL
Caching / Queue	Redis, Celery
Deployment	Railway / AWS / Heroku / Google Cloud

Table 3. Technology stack used in the prototype implementation.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

4.2 Frontend Implementation

The conversational interface supports natural-language product search, displays recommendation cards with price, discount, and rating, renders a side-by-side comparison table for the top candidates, maintains user search history, and supports authentication (sign-up and login). The interface is responsive across desktop, tablet, and mobile form factors.

4.3 Backend and Database

The backend handles incoming queries, coordinates calls to the NLP and Agentic AI modules, manages communication with the scraping/API layer, performs authentication and session management, and persists all conversation, search, and recommendation data. MongoDB or MySQL is used to store user profiles, search history, product records, comparison results, and analytics logs, supporting both fast retrieval and future personalization features.

4.4 Security Implementation

All client–server communication is secured over HTTPS, with JWT-based authentication and role-based access control governing user and admin operations. Passwords are hashed before storage, API keys are managed centrally, and input validation is applied to mitigate injection-style attacks against both the application and the underlying database.

V. RESULTS AND DISCUSSION

5.1 Experimental Setup

Testing was conducted on a development machine with an Intel Core i5 processor, 8 GB RAM, 256 GB SSD storage, and a broadband internet connection, using the software stack listed in Table 3. The chatbot was evaluated using natural-language queries spanning multiple product categories, including smartphones, laptops, headphones, smartwatches, and cameras, with product data retrieved from Amazon, Flipkart, eBay, and Snapdeal.

5.2 NLP Performance

The NLP module was evaluated on its ability to correctly identify intent, extract keywords, recognise named entities, and maintain conversational context. Table 4 summarizes the measured accuracy for each sub-task.

NLP Task	Accuracy
Intent Recognition	94%
Keyword Extraction	93%
Entity Recognition	92%
Conversational Understanding	91%

Table 4. NLP sub-task accuracy.

5.3 Recommendation and Comparison Accuracy

Across the tested queries, the system achieved an overall recommendation accuracy of approximately 92%, with product-data retrieval accuracy of around 91% and a web-scraping success rate of roughly 90%, reflecting the impact of intermittent anti-bot defences on a small fraction of requests. API response success was measured at approximately 94%. For example, given the query "Find the best Samsung mobile under ₹30,000," the system correctly extracted the product category, brand, and budget, retrieved comparable listings from four platforms, and recommended the Flipkart listing at ₹27,499 with a 12% discount and free delivery as the best overall deal, ahead of comparable Amazon and eBay listings.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

5.4 Response Time and System Performance

The end-to-end average response time, from query submission to recommendation display, ranged between 2 and 4 seconds, with average data-retrieval time around 3 seconds per query. The system maintained approximately 99% uptime during testing, with no critical failures observed across functional, integration, and security test cases.

Metric	Result
Average Response Time	2–4 seconds
Recommendation Accuracy	~92%
NLP Accuracy	94%
Product Retrieval Accuracy	~91%
System Uptime	~99%
API Communication Success	~94%

Table 5. Summary of system performance metrics.

5.5 Discussion

These results indicate that combining NLP-based query understanding with an Agentic AI decision layer meaningfully improves both the accuracy and the personalization of deal recommendations relative to static comparison tools. The chatbot consistently identified the correct product, brand, and budget from conversational input, and the Agentic AI layer reliably balanced competing factors (price, rating, delivery speed) rather than defaulting to a single fixed criterion. The principal advantages observed during testing include real-time comparison, reduced manual effort, personalized recommendations, and a responsive, user-friendly interface across device types.

At the same time, several limitations were identified. Web-scraping reliability is sensitive to changes in target-site structure and to anti-bot mechanisms, which can intermittently reduce data-retrieval accuracy. Several e-commerce platforms impose API rate limits that constrain query throughput. Recommendation quality is also bounded by the freshness and completeness of the underlying scraped data, and broader deployment would require additional server resources and ongoing maintenance of the scraping layer as target websites evolve. These limitations motivate the future-work directions discussed in Section 6.

VI. CONCLUSION AND FUTURE WORK

This paper presented the Smart Deal Finder Bot, an Agentic-AI-driven conversational shopping assistant that autonomously compares product prices, discounts, ratings, and delivery options across multiple e-commerce platforms and recommends the most suitable deal through a natural-language interface. By integrating NLP-based query understanding, real-time web scraping and API integration, a weighted multi-parameter comparison engine, and an autonomous Agentic AI decision layer, the system addresses a clear gap in existing price-comparison tools, which generally lack conversational interaction, personalization, and autonomous reasoning. Experimental evaluation demonstrated strong NLP accuracy (94% intent recognition), high recommendation accuracy (approximately 92%), and acceptable response latency (2–4 seconds), confirming the practical feasibility of the proposed approach.

Future work will focus on extending the system along several directions: voice-based conversational interaction using speech-to-text and text-to-speech integration; multilingual support for languages such as Hindi, Marathi, Tamil, French, and Spanish; more advanced personalization through collaborative filtering and deep-learning-based recommendation models trained on purchase history; predictive price analysis to forecast favourable buying windows; a dedicated mobile application with push notifications and real-time alerts; and broader platform coverage including Walmart,

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 10, June 2026
Impact Factor: 8.2

Croma, Meesho, and Ajio. Collectively, these enhancements would further improve the scalability, accessibility, and intelligence of the Smart Deal Finder Bot as a general-purpose autonomous shopping assistant.

REFERENCES

[1] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed. Pearson Education, 2021.
[2] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA: MIT Press, 2016.
[3] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed. Pearson, 2023.
[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in Proc. Int. Conf. Learning Representations (ICLR), 2013.
[5] T. Brown et al., "Language models are few-shot learners," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 1877–1901, 2020.
[6] F. Chollet, Deep Learning with Python, 2nd ed. Shelter Island, NY: Manning Publications, 2021.
[7] M. Grinberg, Flask Web Development: Developing Web Applications with Python. Sebastopol, CA: O'Reilly Media, 2018.
[8] React Documentation. [Online]. Available: https://react.dev
[9] Flask Documentation. [Online]. Available: https://flask.palletsprojects.com
[10] Django Documentation. [Online]. Available: https://www.djangoproject.com
[11] spaCy Documentation. [Online]. Available: https://spacy.io
[12] NLTK Documentation. [Online]. Available: https://www.nltk.org
[13] Beautiful Soup Documentation. [Online]. Available: https://www.crummy.com/software/BeautifulSoup
[14] Scrapy Documentation. [Online]. Available: https://scrapy.org
[15] MongoDB Documentation. [Online]. Available: https://www.mongodb.com
[16] TensorFlow Documentation. [Online]. Available: https://www.tensorflow.org
[17] N. Lahane, A. Patil, M. More, and A. Walke, "Smart Deal Finder Bot Using Agentic AI: An Intelligent Conversational System for Automated E-Commerce Price Discovery and Comparison," Int. J. Advanced Research in Science, Communication and Technology (IJARSCT), vol. 6, no. 6, May 2026