International Journal of Engineering Trends and Applications (IJETA) – Volume 13 Issue 3, May-Jun 2026

RESEARCH ARTICLE OPEN ACCESS

NEST: Voice Enabled Inventory Management System

Dr. Gaurav Kumar Jain*, Hitesh Kumar**, Aakash Bairwa**, Deepak Jangid**, Harsh Pandey**

*Associate Professor, Department of Computer Science and Engineering, Global Institute of Technology, Jaipur
**B.Tech Student, Department of Computer Science and Engineering, Global Institute of Technology, Jaipur

ABSTRACT
Inventory management is a critical function in any organization that deals with physical goods, materials, or assets. Traditional inventory systems rely heavily on manual data entry, which is time-consuming, error-prone, and often inaccessible to non-technical users. This paper presents NEST (Naturalized Efficient Stock Tracker), a voice-enabled inventory management system that integrates advanced speech recognition, natural language processing (NLP), and artificial intelligence (AI) to allow users to perform inventory operations through spoken commands. NEST supports operations such as adding products, updating stock levels, searching inventory, and generating reports all via voice input. The system is built on a cross-platform architecture using Flutter for the mobile frontend, React for the web dashboard, FastAPI as the backend engine, and MySQL for persistent data storage. AI modules provide intelligent command interpretation, predictive stock alerts, and smart recommendations. Designed for small and medium-scale enterprises (SMEs), NEST reduces operational complexity, improves accuracy, and enhances accessibility, delivering a next-generation approach to inventory management.

Keywords — Inventory Management, Voice Recognition, Natural Language Processing, Artificial Intelligence, Flutter, FastAPI, Speech-to-Text, Stock Tracking, Mobile Application.

1. INTRODUCTION

Inventory management encompasses the tracking, organization, and control of goods within a supply chain or warehouse. Efficient inventory control ensures that the right products are available in the right quantities at the right time, preventing both stockouts and overstocking two major sources of financial loss in businesses of all sizes. Conventional inventory systems have progressed from handwritten ledgers to spreadsheet tools and barcode-based digital platforms. However, even modern systems largely require keyboard input and screen-based interaction, posing challenges in environments where hands-free operation is necessary, or where users have limited technical literacy.

Voice-enabled interfaces have emerged as a transformative technology across various domains including healthcare, automotive, and consumer electronics. The integration of speech recognition with business applications has the potential to significantly enhance productivity and accessibility. Studies indicate that voice interfaces can improve task completion speed by up to 30% in operational settings by eliminating the need for typing and navigating complex menus [1].

This paper introduces NEST (Naturalized Efficient Stock Tracker), a voice-enabled inventory management system designed to address the limitations of existing approaches. NEST allows users to interact with their inventory using natural spoken commands, backed by AI-driven interpretation and a robust, scalable backend architecture. The system targets small and medium enterprises that require a cost-effective, easy-to-use, and intelligent stock management solution.

The key contributions of this work are:

- A complete architecture for a voice-first inventory management system.
- Integration of AI modules for command interpretation, error correction, and predictive analytics.
- A cross-platform implementation covering mobile (Flutter), web (React), and backend (FastAPI/MySQL).
- Evaluation of system usability and accuracy through real-world scenario testing.

2. LITERATURE REVIEW

Inventory management systems have undergone significant transformation over the past few decades. Early digital solutions focused on automating the record-keeping aspect of inventory through spreadsheets and simple database applications. The introduction of barcode scanning and RFID technology in the 1990s and 2000s marked a significant leap, allowing for automated data capture during stock movements [2]. Enterprise Resource Planning (ERP) systems subsequently integrated inventory management with other business functions such as procurement, sales, and accounting. However, ERP solutions remain prohibitively complex and expensive for small and medium enterprises, creating a gap that simpler, more accessible tools could fill.

The emergence of voice recognition technology, particularly with the advent of deep learning-based models such as DeepSpeech and Whisper, has opened new possibilities for hands-free system interaction. Google Cloud Speech-to-Text and Microsoft Azure Cognitive Services Speech API now provide high-accuracy, real-time transcription capable of handling natural language queries in noisy environments [3][4]. Natural Language Processing (NLP) has evolved from rule-based systems to transformer-based models that can understand context, intent, and semantic meaning with high precision. This enables systems to interpret varied phrasings of the same command for example, 'add 50 units of sugar' and 'update sugar quantity to 50' can be resolved to the same inventory operation [5]. Recent literature has also explored AI-driven predictive analytics in inventory management, where machine learning models anticipate stockout events based on historical consumption patterns and seasonal trends. Such proactive alerts reduce both overstocking costs and supply disruption risks [6].

Compared to existing solutions, NEST uniquely combines all three pillars voice input, NLP command interpretation, and predictive AI analytics within a single, lightweight, cross-platform application suited for SMEs.

3. SYSTEM ARCHITECTURE

The NEST system follows a layered, client-server architecture comprising a frontend presentation layer, a backend processing layer, a database layer, and an AI/voice processing layer. Figure 1 illustrates the high-level system architecture.

3.1 Frontend Layer

The frontend is implemented using two technologies to cover different access scenarios:

- Flutter (Mobile Application): Flutter enables the development of a single codebase that compiles natively to Android and iOS. The mobile application serves as the primary interface for warehouse staff, featuring a prominent voice input button, real-time stock dashboards, and push notification support for low-stock alerts.
- React (Web Dashboard): A React-based web interface provides a comprehensive admin panel for managers, supporting inventory analytics, report generation, user management, and detailed stock views with charts and data tables.

Both interfaces are designed with a responsive, accessible layout, ensuring usability for non-technical users.

3.2 Backend Layer

The backend is powered by FastAPI, a high-performance Python web framework built on ASGI, which enables asynchronous request handling, automatic API documentation (via OpenAPI/Swagger), and excellent scalability. The backend responsibilities include:

- Routing and processing user requests from both frontend clients.
- Executing business logic for inventory operations (CRUD — Create, Read, Update, Delete).
- Handling user authentication and role-based authorization (Admin, Manager, Staff).
- Interfacing with the database layer via SQL queries.
- Communicating with voice recognition APIs and AI processing modules.

3.3 Database Layer

MySQL serves as the relational database management system. It stores:

- Product/inventory records (name, SKU, category, quantity, unit price, supplier).
- Transaction logs (stock additions, removals, adjustments with timestamps and user IDs).
- User account and role data.
- System configuration and alert thresholds.

MySQL's ACID compliance ensures data integrity during concurrent operations, while its indexing capabilities support fast query performance on large inventory datasets.

3.4 Voice and AI Processing Layer

Voice input from the user is captured via the device microphone. The audio stream is forwarded to a

Speech-to-Text API (Google Cloud Speech-to-Text or equivalent), which returns the transcribed text. This text is then processed by an NLP module that:

- Tokenizes and parses the command.
- Identifies the intent (add, update, search, delete, report).
- Extracts entities (product name, quantity, category, date).
- Maps the interpreted command to a backend API call.

AI modules further provide smart inventory suggestions, predictive stock-out alerts based on consumption trends, and intelligent error correction for misheard commands.

4. METHODOLOGY

The development of NEST follows an iterative, agile methodology with clearly defined processing stages for each user interaction. The system workflow proceeds as follows:

Step 1: Voice Input Capture

The user activates the microphone via the voice command button on either the mobile or web interface. Audio is captured in real time through the device microphone hardware API.

Step 2: Speech-to-Text Conversion

The captured audio stream is transmitted to the speech recognition engine, which converts the spoken input into a text transcript. The system supports continuous speech and handles natural pauses to determine the end of a command.

Step 3: NLP Command Interpretation

The transcribed text is analyzed by the NLP pipeline. A rule-based intent classifier identifies the primary action requested by the user. Named Entity Recognition (NER) extracts relevant parameters such as product name and quantity. The system supports a broad range of natural language phrasings to maximize usability.

Step 4: Backend Processing

The interpreted command, along with extracted parameters, is sent to the FastAPI backend as a structured API request. The backend validates inputs, enforces authorization rules, and executes the corresponding database operation.

Step 5: Database Interaction

MySQL processes the SQL query generated by the backend fetching current stock levels, inserting new records, updating quantities, or retrieving report data. All transactions are logged with timestamp and user identity for audit purposes.

Step 6: Response and Feedback

The system returns a confirmation response to the frontend, which is displayed visually (e.g., updated inventory table) and optionally delivered as voice feedback through text-to-speech synthesis. Error messages are clearly communicated to guide the user in re-attempting failed commands.

Step 7: Real-Time Monitoring and Alerts

A background monitoring service continuously evaluates stock levels against predefined thresholds. When any item's quantity falls below its minimum threshold, the system generates an automatic alert notification, displayed on the dashboard and sent as a push notification to authorized users.

5. IMPLEMENTATION

The NEST application was implemented and tested across mobile (Android) and web platforms. The Android application package (APK) was compiled using Flutter and tested on devices running Android 8.0 and above. Key implementation details include:

5.1 Technology Stack

Table 1: NEST Technology Stack

Component	Technology	Purpose
Mobile Frontend	Flutter (Dart)	Cross-platform mobile UI
Web Frontend	React (JavaScript)	Admin dashboard & analytics
Backend API	FastAPI (Python)	Request handling, business logic
Database	MySQL	Persistent data storage
Voice Input	Speech-to-Text API	Audio-to-text transcription
NLP / AI	Python NLP Modules	Command interpretation, prediction

5.2 Voice Command Examples

The following table illustrates representative voice commands supported by NEST and their corresponding system actions:

Table 2: Sample Voice Commands and Actions

Voice Command	System Action
"Add 100 units of rice"	Add product / update quantity to 100
"How many bags of sugar are left?"	Query current stock level for sugar
"Update price of wheat flour to 45 rupees"	Update product unit price
"Generate stock report for this month"	Generate monthly inventory report
"Delete product cement from inventory"	Remove product record
"Show items below minimum stock"	Display low-stock alert list

6. RESULTS AND DISCUSSION

The NEST system was evaluated through functional testing across multiple inventory operation scenarios. Testing was conducted on both the Android application and the web dashboard. The following observations were recorded:

- Voice Recognition Accuracy: The system achieved a command recognition accuracy of approximately 91% in a quiet environment. In moderate ambient noise conditions, accuracy remained above 83%, which is acceptable for typical warehouse or store settings.
- Command Interpretation: The NLP pipeline correctly resolved user intent in 88% of test cases on the first attempt. Unrecognized commands were gracefully handled with clear error prompts asking the user to rephrase.
- Response Latency: Average end-to-end response time from voice input to confirmed system action was 1.8 seconds for standard operations (add/update/search), with report generation taking up to 4.2 seconds depending on dataset size.
- Cross-Platform Consistency: The Flutter mobile app and React web dashboard maintained visual and functional consistency, with both reflecting inventory updates in near real-time (under 2 seconds).
- Low-Stock Alerts: The automated alert system correctly triggered notifications for all items that fell below threshold during testing, with zero false negatives observed.

These results demonstrate that NEST provides a practical and efficient voice-first inventory management solution. The system compares favourably to conventional keyboard-driven interfaces in terms of speed of data entry and accessibility, particularly for users with limited typing proficiency.

7. FUTURE SCOPE

While NEST achieves its core objectives effectively, several enhancements are planned for future development:

- Multilingual Voice Support: Extending voice command recognition to support regional Indian languages including Hindi, Rajasthani, and other vernaculars, improving accessibility for a wider user base.
- Advanced Predictive Analytics: Integrating machine learning models trained on historical transaction data to forecast future stock requirements, seasonal demand spikes, and optimal reorder points.
- Barcode and QR Code Integration: Combining voice input with barcode/QR scanning for hybrid data entry, suitable for environments requiring both modes.
- Cloud Deployment and Multi-Tenancy: Deploying NEST on cloud infrastructure (AWS/GCP/Azure) to support multiple organizations (multi-tenant architecture) with data isolation, scalability, and high availability.
- IoT Integration: Connecting NEST with IoT-enabled smart shelves or RFID readers for fully automated, sensor-driven inventory tracking.
- Offline Mode: Implementing local data caching to allow inventory operations in offline or low-

connectivity environments, with automatic synchronization when connectivity is restored.

8. CONCLUSION

This paper presented NEST (Naturalized Efficient Stock Tracker), a voice-enabled inventory management system that leverages speech recognition, natural language processing, and artificial intelligence to modernize stock management operations. By replacing conventional keyboard-centric interfaces with an intuitive voice-first interaction model, NEST significantly reduces the cognitive and physical effort required to manage inventory. The system's layered architecture combining Flutter for mobile, React for web, FastAPI for backend services, and MySQL for data persistence ensures scalability, maintainability, and cross-platform accessibility. AI-powered features such as smart command interpretation, predictive alerts, and intelligent error correction further distinguish NEST from conventional inventory tools. NEST is particularly well-suited for small and medium enterprises seeking a cost-effective, user-friendly, and intelligent inventory solution. The results of functional testing confirm the system's viability and effectiveness. With planned enhancements including multilingual support, predictive analytics, and IoT integration, NEST is positioned to become a comprehensive next-generation inventory management platform.

ACKNOWLEDGEMENT

The authors extend their sincere gratitude to the Department of Computer Science & Engineering, Global Institute of Technology, Jaipur, for providing the resources, infrastructure, and academic environment that made this project possible. We also thank our faculty mentors for their invaluable guidance throughout the development of NEST.

REFERENCES

[1] S. Ruan, J. O. Wobbrock, K. Liou, A. Ng, and J. A. Landay, “Comparing speech and keyboard text entry for short messages in two languages on touchscreen phones,” Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 1, no. 4, pp. 1–23, 2018.

[2] K. H. Lai, E. W. T. Ngai, and T. C. E. Cheng, “Measures for evaluating supply chain performance in transport logistics,” Transportation Research Part E: Logistics and Transportation Review, vol. 38, no. 6, pp. 439–456, 2002.

[3] Google Cloud, “Speech-to-Text API Documentation,” Google Cloud, 2024. Available: https://cloud.google.com/speech-to-text

[4] Microsoft, “Azure Cognitive Services Speech Service,” Microsoft Learn, 2024. Available: https://learn.microsoft.com/azure/ai-services/speech-service/

[5] C. D. Manning and H. Schütze, Foundations of Statistical Natural Language Processing. Cambridge, MA, USA: MIT Press, 1999. Available: https://nlp.stanford.edu/

[6] IBM, “Artificial Intelligence in Business Applications,” IBM, 2024. Available: https://www.ibm.com/topics/artificial-intelligence

[7] Flutter Development Team, “Flutter Framework Documentation,” Flutter, 2024. Available: https://docs.flutter.dev/

[8] Meta Open Source, “React Official Documentation,” React, 2024. Available: https://react.dev/

[9] S. Ramirez, “FastAPI Official Documentation,” FastAPI, 2024. Available: https://fastapi.tiangolo.com/

[10] Oracle Corporation, “MySQL 8.0 Reference Manual,” Oracle, 2024. Available: https://dev.mysql.com/doc/

[11] Investopedia, “Inventory Management: Definition, How It Works, Methods & Examples,” Investopedia, 2024. Available: https://www.investopedia.com/terms/i/inventory-management.asp

[12] Aayushi, L. Yadav, P. Paliwal, A. Johari, R. Ajmera, and G. K. Soni, “A Simulation-Based Evaluation of Machine Learning Models for Algorithmic Trading in Equity Markets,” in Proc. 6th Int. Conf. Expert Clouds and Applications (ICOECA), pp. 1049–1054, 2026.

[13] H. Sharma and R. Ajmera, “Comprehensive Review and Analysis of Elderly Fall Detection System Using Machine Learning,” Tuijin Jishu/Journal of Propulsion Technology, vol. 44, no. 5, 2023.

[14] K. Ahuja, H. Sekhawat, S. Mishra, and P. Jha, “Machine Learning in Artificial Intelligence:

Towards a Common Understanding," Turkish Online Journal of Qualitative Inquiry, vol. 12, no. 8, 2021.

[15] K. Paliwal, P. Jha, S. Kumari, V. Vaish, N. Vishwakarma, and A. Bansal, "Machine Learning in Electric Vehicle Consumption Modelling," in Proc. 9th Int. Conf. Intelligent Computing and Control Systems (ICICCS), pp. 727–730, 2026.

[16] P. Jha, P. Jain, A. Kumar, S. Soni, Y. Sharma, and P. Agarwal, "The Application of Markov Chains to Linguistic Predictions by Utilising Its Inherent Information Entropy," in Proc. 9th Int. Conf. Inventive Systems and Control (ICISC), pp. 1110–1114, 2025.

[17] P. Jha, T. Biswas, U. Sagar, and K. Ahuja, "Prediction with ML Paradigm in Healthcare System," in Proc. 2nd Int. Conf. Electronics and Sustainable Communication Systems (ICESC), pp. 1334–1342, 2021.

[18] A. Johari, R. Sharma, A. Meena, and V. Tiwari, "Advancements in Pre-Trained Language Models and Their Impact on Various NLP Tasks," International Journal of Engineering Trends and Applications, vol. 11, no. 3, pp. 201–209, 2024.

[19] K. Gautam, G. K. Soni, R. Ajmera, N. Hemrajani, J. Ahuja, and M. K. Jha, "Deep Reinforcement Learning for Stock Market Portfolio Optimization," in Proceedings of the 5th International Conference on Communication, Computing and Electronics Systems (ICCCES), pp. 1835–1839, 2026.

[20] M. K. Jha, G. K. Soni, G. Jain, S. Tiwari, K. Gupta, and B. Singhal, "Comparative Analysis of Classical Machine Learning Models for Twitter Sentiment Classification," in Proceedings of the 5th International Conference on Communication, Computing and Electronics Systems (ICCCES), pp. 1949–1954, 2026.

[21] N. Bhargava and A. Johari, "Enhancing deep learning approach for Tamil-English mixed text classification," in Proceedings of the International Conference on Applications of Machine Intelligence and Data Analytics (ICAMIDA 2022), Advances in Computer Science Research, pp. 829–837, 2023.

[22] A. Jangir, A. Agrawal, C. Sharma, G. K. Soni, R. Ajmera, and A. Johari, "Comparative Performance Analysis of Deep Learning and Traditional Algorithms for Facial Recognition and Image Classification," in Proceedings of the 4th International Conference on Automation, Computing and Renewable Systems (ICACRS), pp. 1172–1175, 2025.

[23] A. Maheshwari and R. Ajmera, "A comprehensive guide to natural language processing in Sanskrit with named entity recognition," in Proceedings of the ACM International Conference on Information Management and Machine Intelligence (ICIMI), 2023.

[24] R. Misra and N. Sharma, "Artificial Intelligence Driven Cybersecurity Techniques: Challenges and Future Directions," International Journal of Engineering Trends and Applications, vol. 13, no. 1, pp. 11–16, 2026.

[25] S. A. Saiyed, N. Sharma, H. Kaushik, P. Jain, G. K. Soni, and R. Joshi, "Transforming Portfolio Management with AI and ML: Shaping Investor Perceptions and the Future of the Indian Investment Sector," in Proceedings of the Parul University International Conference on Engineering and Technology (PiCET), pp. 1108–1114, 2025.

[26] S. Thapar, G. K. Soni, H. Kaushik, R. Singh, S. Bisht, and S. K. Bansal, "A Comparative Machine Learning Framework for Detecting Fake Accounts on Facebook," in Proceedings of the 4th International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), pp. 1567–1571, 2025.

[27] N. Sharma, "A Study on Artificial Intelligence Applications in Autonomous Vehicle Systems," International Journal of Engineering Trends and Applications, vol. 13, no. 2, pp. 11–14, 2026.

[28] R. Ajmera, A. Johari, A. Goyal, A. Purohit, A. Kumar, and J. A. Ashok, "Multilingual Sentiment Analysis Based on Fine-Tuned Transformer Architectures," in Proceedings of the 5th International Conference on Communication, Computing and Electronics Systems (ICCCES), pp. 1589–1592, 2026.

[29] M. K. Jha, K. Kumar, N. Hemrajani, D. S. Rao, A. Goyal, and R. Ajmera, "AI Powered Student Performance Prediction using Explainable ML," in Proceedings of the 4th International Conference on Automation, Computing and Renewable Systems (ICACRS), pp. 1140–1144, 2025.

[30] I. Yadav, V. Shekhawat, K. Gautam, G. K. Soni, and R. Yadav, "Artificial Intelligence for Cybersecurity: Emerging Techniques, Challenges, and Future Trends," in Proceedings of the 3rd International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), pp. 1176–1180, 2025.

[31] M. K. Jha, "Recent Trends and Emerging Applications of the Internet of Things: Transforming the Way We Live and Work," International Journal of Engineering Trends and Applications, vol. 12, no. 4, pp. 239–244, 2025.

[32] M. K. Jha, S. Agarwal, and V. Kabra, "Artificial Intelligence at Work: Transforming Industries and Redefining the Workforce Landscape," International Journal of Engineering Trends and Applications, vol. 12, no. 4, pp. 416–424, 2025.

[33] S. Kumari, S. Sharma, A. Johari, G. K. Soni, H. Kaushik, and S. Jain, "Twitter Sentiment Analysis using Machine Learning Techniques," in Proceedings of the 6th International Conference on Expert Clouds and Applications (ICOECA), pp. 1711–1717, 2026.