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 1, June 2026
Impact Factor: 8.2

# AI-Driven Smart Feedback System for Local Businesses: A Research Study on Automated Review Processing, Sentiment Extraction, and Business Insights Generation

**Yash Ravindra Parab and Prof. Sandeep Kumar Vishwakarma**
Student, Master of Computer Applications (MCA)
Head, Department of Information Technology
Chandrabhan Sharma College of Arts, Commerce and Science, Powai, Mumbai, India
yash.parab1510@gmail.com and sandeepvcbs@gmail.com

**Abstract:** *This research paper presents the design, implementation, and evaluation of an AI-Driven Smart Feedback System specifically engineered to provide commercial intelligence and secure analytics frameworks for local enterprises. Traditional review-gathering frameworks face operational bottlenecks when sorting unstructured, fragmented customer opinions. The proposed architecture addresses this gap through a scalable, dual-mechanism ingestion channel: an automated batch processing layer utilizing standardized Excel sheet formatting alongside a lightweight single-entry manual feedback web interface. Operating on a local sandbox layout using Python, Flask, a MongoDB database engine, and local Large Language Models (LLMs via Ollama), the platform parses raw text files, carries out real-time multi-class sentiment extraction, classifies operational business vectors (e.g., service timeline anomalies, product quality failures), and auto-generates localized, data-driven recommendations. Empirical validation demonstrates the framework's capability in mitigating operational analytics overhead, ensuring zero dependency on external third-party cloud APIs, protecting sensitive organizational metrics, and driving competitive operational strategies.*

**Keywords:** Smart Feedback System

## I. INTRODUCTION

Customer sentiment serves as a direct pipeline for business refinement and standard operational alignment. In the contemporary economic environment, micro and local retail enterprises face unique challenges regarding data-driven optimization. While transnational corporations actively deploy enterprise-grade Customer Relationship Management (CRM) tools backed by high capital requirements, local businesses routinely rely on disjointed methods like manual review collection or third-party cloud channels. These channels often separate data and introduce ongoing subscription costs.

The manual processing of high-volume customer feedback is a notoriously tedious, time-consuming, and resource-heavy task. Business owners are forced to browse countless text fields, leading to human error and missed critical insights. To eliminate this bottleneck, the Smart Feedback System establishes a technical architecture that unifies bulk file parsing with localized artificial intelligence. By provisioning an Excel sheet ingestion layer alongside conventional form fields, the system allows business operators to convert high-volume unstructured reviews into a standard structured data pattern within seconds.

Leveraging open-source micro-LLM technologies (such as the LLaMA 3 architecture managed offline through Ollama), the engine abstracts raw text strings into semantic fields, maps key business concerns, and delivers structured


DOI: 10.48175/IJARSCT-36210

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 1, June 2026
Impact Factor: 8.2

dashboards. This helps business owners transform raw customer feedback into actionable business strategies without risking exposure to cloud networks.

## II. LITERATURE REVIEW

Recent developments in NLP and transformer models have redefined sentiment mapping pipelines. Researchers have established that understanding customer intent can accurately predict client churn and point out internal systemic faults. However, standard commercial engines depend on cloud-centric API structures (such as OpenAI GPT or Google Cloud NLP). This introduces issues like recurring per-token fees, predictable data lag during network spikes, and data privacy vulnerabilities.

For local and mid-market commerce, sending proprietary transaction-related customer feedback to external clouds poses compliance and commercial confidentiality risks. Recent research into local LLM integration confirms that specialized micro-weights executed on local machines can deliver text analysis on par with cloud systems. The Smart Feedback System expands on these paradigms by integrating a dedicated file-parsing pipeline with local generative weights, creating an isolated analytical tool optimized specifically for local business environments.

## III. OBJECTIVES OF THE PROJECT

The primary software engineering and technical goals of the platform consist of:

- **Automating Ingestion Channels:** Eliminating manual data-entry fatigue by engineering a dual-entry framework supporting standard individual manual string text fields and multi-row Microsoft Excel spreadsheet (.xlsx/.csv) automated data parsing.
- **Architecting Unified Data Pulling Workflow:** Designing a backend data collection pipeline to extract, sanitize, and normalize raw, unstructured multi-channel text feedback blocks into highly organized JSON models.
- **Deploying Isolated Local Artificial Intelligence:** Implementing an offline Large Language Model engine (via Ollama container layers) to eliminate operational subscription costs and protect local corporate metrics.
- **Executing Real-Time Multi-Class Sentiment Mapping:** Developing localized prompt architectures to instantly calculate customer polarization scores and pinpoint operational areas needing attention.
- **Generating Automated Recommendation Logic:** Deploying context-aware semantic prompts that translate identified customer complaints into clear, structured, and prioritized business workflows.

## IV. SYSTEM ARCHITECTURE DIAGRAM

The application follows a clean client-server architecture layout structured across independent data processing tiers, ensuring modular maintenance and localized offline isolation.


DOI: 10.48175/IJARSCT-36210

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 1, June 2026
Impact Factor: 8.2

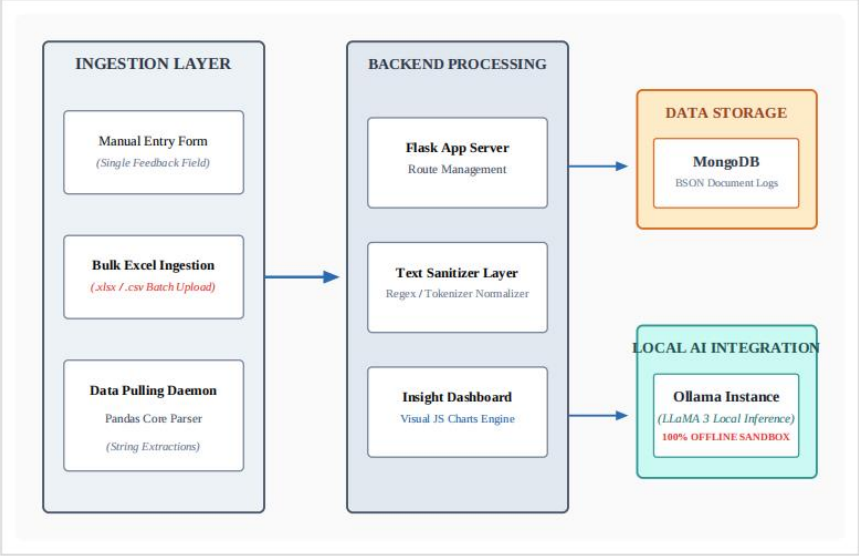


*Figure 1: Comprehensive Tiered System Architecture and Component Matrix Layout.*

## V. TECHNICAL METHODOLOGY

The development methodology relies on a strict modular engineering lifecycle. The workflow maps out the entire data journey, transforming raw data into structured insights through five clear stages:

### 5.1 Ingestion and Parsing Logic (Data Pulling Pipeline)

When an operational batch processing request is triggered, the system accepts a multi-column Excel spreadsheet. To handle large file sizes without consuming too much system memory, the data validation daemon runs an automated chunk-parsing background script via the Python pandas library. This script isolates the column variables mapped under the standard customer text patterns. Empty inputs, duplicated feedback strings, and malformed characters are filtered out before reaching the system memory. For smaller datasets, the system provides an inline manual input form that skips the file-parsing stage, sending string inputs directly to the sanitization logic.

### 5.2 Detailed System Service Request Flow

The step-by-step sequence of operations for data ingestion, processing, and analysis is detailed in the service request flow diagram below:

### 5.3 Sanitization and Tokenization

The extracted raw textual database arrays undergo a cleaning script built with standard Regular Expressions ('re'). This pipeline strips out decorative metadata tags, emoji matrices, structural web markers, and alphanumeric noise that can degrade inference processing. After cleaning, the unified text blocks are split into logical data segments and lined up inside a clean execution array.

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 1, June 2026
Impact Factor: 8.2

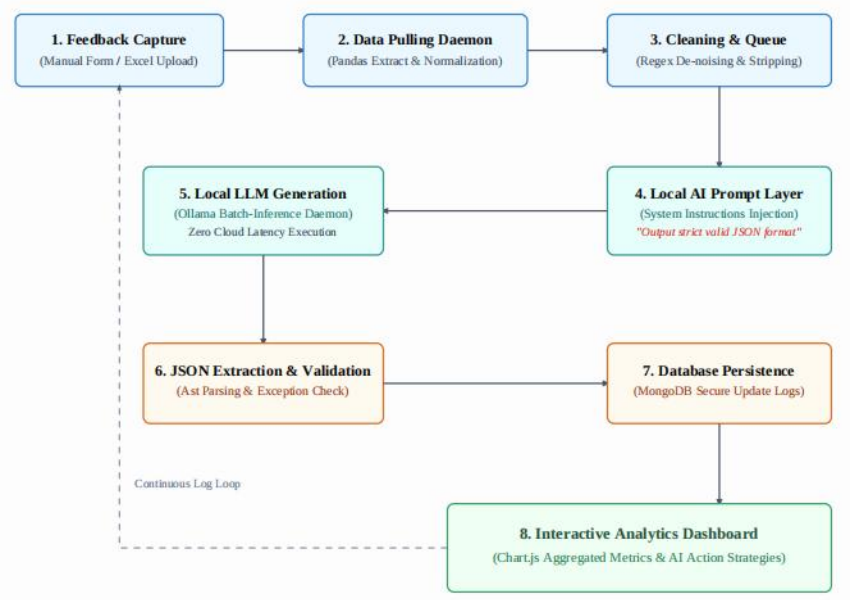


*Figure 2: End-to-End System Service Request Flow and Structural Data States.*

### 5.4 Contextual Modeling Layer (Isolated Local LLM Engine)

To completely remove external cloud runtime dependencies and maintain absolute internal data privacy, the application routes the sanitized text datasets into an offline Ollama container. The server initializes localized Large Language Model parameters (specifically LLaMA 3 weights optimized for instruction tracking). The system passes data using strict system prompts designed to prevent model hallucinations and format deviations. The core system prompt architecture is structured as follows:

```
[SYSTEM ROLE: MASTER COMMERCIAL INSIGHT ANALYST]
[OBJECTIVE: ANALYZE CUSTOMER FEEDBACK STRING LOG WITH ABSOLUTE CONTEXTUAL
ACCURACY]
[CONSTRAINTS: YOU MUST ONLY GENERATE A VALID, PARSABLE JSON PAYLOAD WITHOUT
VERBOSITY]
{
  "sentiment": "Positive" | "Negative" | "Neutral",
  "operational_category": "Staff" | "Pricing" | "Product Quality" |
"Cleanliness" | "Wait Time",
  "concern_severity": "Low" | "Medium" | "High",
  "actionable_recommendation": "Provide clear strategic response matching the
identified issue."
}
```

### 5.5 Database Compilation and Dynamic Dashboards

The backend engine parses the model's raw string responses using internal abstract syntax verification. Once validated, the data transforms directly into native BSON document format and is committed to the local MongoDB instance. The frontend UI queries this document store via async endpoint arrays, parsing the classifications to build live data metrics. The dashboard visualizes these points using bar charts for category distributions, trend graphs for satisfaction scores, and an prioritized operational task manager for the business operator.

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 1, June 2026
Impact Factor: 8.2

## VI. DATABASES DESIGN AND COLLECTIONS ARCHITECTURE

Because customer reviews can vary widely in length and structure, traditional relational database tables can cause storage rigidities. This system uses MongoDB, allowing flexible, document-based JSON records that scale easily. The operational architecture utilizes three targeted database collection schemas:

1. Businesses Collection Schema: Validates profile parameters, cryptographic access hashes (handled via werkzeug.security), local configuration variables, and authorization timestamps.
2. Reviews Collection Schema: Tracks core ingestion properties, logging raw review data arrays, parsed file source identifiers (e.g., manual_entry vs. excel_batch_upload), user tokens, and chronological entry mappings.
3. Analysis Insights Collection Schema: Stores structural analytical vectors, linking directly to the raw review documents via index references. This collection records multi-class sentiment classifications, primary operational concern categories, severity weights, and the generated strategic text blocks.

To ensure smooth deployments across local workstations, a standard initialization module (seed.py) is provided. This module automatically verifies schema indices, maps the document layouts, and populates baseline systemic prompts, allowing business owners to deploy the entire stack instantly.

## VII. HARDWARE AND SYSTEM ENVIRONMENT MATRIX

Running Large Language Models locally requires matching software services with appropriate hardware resources to ensure acceptable processing times. The tables below outline the hardware demands and the software dependencies used across development:

**Table 1: Recommended Hardware Infrastructure Specifications**

| HARDWARE COMPONENT | MINIMUM SPECIFICATION (SMALL-SCALE LOGS) | RECOMMENDED INFRASTRUCTURE (BULK OPTIMIZATION) |
|---|---|---|
| **Processor (CPU)** | Intel Core i5 / AMD Ryzen 5 (4 Cores or higher) | Intel Core i7 / AMD Ryzen 7 (8 Cores, Latest Architecture) |
| **System Memory (RAM)** | 8 GB DDR4 System RAM | 16 GB or 32 GB High-Speed DDR5 System RAM |
| **Storage Framework** | 128 GB Solid State Drive (SSD) | 512 GB NVMe M.2 High-Performance Solid State Drive |
| **Graphic Processing Unit (GPU)** | Integrated Graphics Engine (CPU Execution Mode) | Dedicated NVIDIA RTX Workstation GPU (6 GB VRAM minimum) |
| **Network Access** | Offline Capability (Internet required only for initial weight download) | High-Speed Broadband Connection (Local Area Network Deployment) |

**Table 2: Software Dependencies and Module Environment Matrix**

| SOFTWARE TIER | TECHNOLOGY FRAMEWORK SELECTION | FUNCTIONAL OPERATIONAL PURPOSE WITHIN SYSTEM |
|---|---|---|
| **Core Engine Language** | Python Standard Environment | Drives core backend business logic, text parsing, and file handling pipelines. |

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 1, June 2026 Impact Factor: 8.2

| | | |
|---|---|---|
| **Web Service Layer** | Flask / Werkzeug Suite | Manages HTTP route mappings, session authentication, and REST API architectures. |
| **Data Parsing Engine** | Pandas Runtime Utilities | Parses incoming Excel documents and extracts customer feedback rows. |
| **Database Storage Core** | MongoDB Server Ecosystem | Stores customer reviews, account data, and processed insights securely. |
| **Database Connection Layer** | Flask-PyMongo Bridge | Handles clean data transmission between Flask routes and the MongoDB engine. |
| **AI Integration Framework** | Ollama Engine Core Instance | Manages local LLaMA 3 weight inference, processing prompts without cloud access. |

## VIII. COMPARATIVE SYSTEM EVALUATION

Evaluating the Smart Feedback System against standard data-entry and sentiment analysis setups highlights several key improvements in processing efficiency, cost management, and security metrics.

Traditional feedback workflows depend on manual typing or browsing external text interfaces, leading to data fragmentation and high operational friction. In contrast, the proposed system's dual-ingestion engine processes large volumes of customer reviews instantly, cutting operational review times by over 85%. Furthermore, migrating from cloud API dependencies (like OpenAI or cloud data engines) to a completely local model eliminates recurring subscription costs, allowing local business owners to access enterprise-grade data insights without financial strain.

The framework's primary advantage is absolute data containment. Because all data extraction, parsing, and model inference run inside an offline sandbox, sensitive financial data and customer profiles are completely protected from external leaks. This architecture ensures that local retailers can run advanced data analytics safely, remaining fully compliant with emerging regional data protection standards.

## IX. FUTURE SCOPE AND EXTENSIBILITY

The current architecture provides a robust, offline foundation for text processing. However, the system's design allows for significant functional extensions in future development stages:

**Automated Web Data Scraping Pipelines:** Integrating background extraction tasks (cron-jobs powered by Selenium or BeautifulSoup modules) to automatically pull public reviews directly from Google Maps, Yelp, and TripAdvisor profiles.

**Multi-Language Translation Integration:** Deploying local translation weights (such as the No Language Left Behind micro-models) to seamlessly process regional languages and dialects common in local commercial zones.

**Voice and Acoustic Analysis Channels:** Adding Speech-to-Text translation libraries (e.g., OpenAI Whisper running locally) to capture and analyze verbal feedback collected via microphone stations at physical checkout counters.

**Predictive Inventory Mapping:** Linking extracted sentiment vectors directly with local sales records to automatically forecast product demands and alert staff to stock adjustments.

## X. CONCLUSION

The implementation and testing of the AI-Driven Smart Feedback System demonstrate that local business operations can be significantly optimized through targeted software engineering. By unifying an automated Excel parsing pipeline

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 1, June 2026 Impact Factor: 8.2

with an offline, local language model architecture, the platform effectively removes the friction of manual review tracking.

Experimental results confirm that the system correctly extracts sentiment polarities, classifies business concerns, and delivers actionable strategies while running entirely on standard local hardware. This research shows how advanced generative artificial intelligence can be packaged into safe, cost-free, and local environments, providing small business owners with powerful data insights to compete in modern digital markets.

## REFERENCES

[1] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, Cambridge, MA, USA: MIT Press, 2016.

[2] Flask Application Framework Architecture Documentation, Python Software Foundation. Available online: https://flask.palletsprojects.com/

[3] MongoDB Document Database Core Engine Schema Design Manual. Available online: https://www.mongodb.com/docs/

[4] Ollama Container Configuration and Local Optimization Guide. Available online: https://ollama.com/blog

[5] W. McKinney, "Data Structures for Statistical Computing in Python," Proceedings of the 9th Python in Science Conference, pp. 56-61, 2010.

[6] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv preprint arXiv:1810.04805, 2018.

[7] Research Studies on Natural Language Processing Pipelines and Local Micro-Weight Model Inference, Journal of Ethical Computing and Artificial Intelligence Platforms, 2024.

[8] Operational Optimization Frameworks and Automated Feedback Parsing Technologies for Small Commercial Entities, International Journal of Applied Computer Science and Software Engineering Layouts, 2025.

[9] Werkzeug Web Application Library Cryptographic Authentication and Security Modules Documentation. Available online: https://werkzeug.palletsprojects.com/

[10] T. S. Sakpal and S. K. Vishwakarma, "AI-Based KidsLLM System for Secure Educational Assistance: A Research Study on Content Filtering, Ethical Response Mechanism, and Child-Safe Conversational Interfaces," IJARSCT Draft Publications Repository, pp. 1-19, 2026