Joolca Customer Support RAG System

Final Internship Project Report
Ishaan Gosain

Executive Summary

Developed an automated customer support system for Joolca using Retrieval-Augmented Generation (RAG) technology. The system transforms historical support tickets into a searchable knowledge base, answers customer queries with cited sources, and escalates unresolved or complex issues to human agents. Built using n8n workflow automation, Supabase vector database, OpenAI/Anthropic language models, and Zendesk.

Key Results:

- Processed 1000+ historical support tickets into a vector-searchable knowledge base
- Created automated response pipeline with citation of ticket IDs
- Implemented seamless escalation workflow with Zendesk integration
- Delivered a fully working, documented system ready for production

---

1. Project Overview

Problem

Joolca's support team previously had to manually search historic tickets, resulting in slow responses and inconsistent customer experiences.

Solution

Designed and implemented an automated RAG system that:

- Searches past support tickets for similar issues using natural-language queries
- Synthesizes multi-chunk answers with citation of source tickets

- Escalates “unknown” queries to human specialists, pre-filled with relevant research/context

Technologies Used

- Orchestration: n8n (35+ connected nodes)
- Database: Supabase + PostgreSQL vector extension
- AI Models: OpenAI GPT-4, Anthropic Claude 3.5 Haiku
- Integrations: Zendesk API, Perplexity API, Gmail API

---

2. Technical Implementation

2.1 Data Processing

- Downloaded historical ticket and comment CSVs from GitHub.
- Built custom JavaScript CSV parsers to handle quoted, multiline fields.
- Combined ticket data with all associated support comments.
- Generated structured records rich in metadata (product, region, status, etc.).

2.2 Vector Database Setup

- `support_knowledge` table stores all chunks with relevant metadata and vector embedding.
- Text split into ~350-character overlapping chunks for retrieval accuracy.
- Vector embeddings generated using OpenAI’s `text-embedding-3-small` model.
- Vectors indexed for fast nearest-neighbor search.

2.3 Workflow Architecture

- Input processing: Captures and parses customer query/context.
- Vector search: Semantic retrieval of top-matching ticket chunks.
- Response generation: LLM synthesizes answer, citing all source ticket IDs.
- Escalation path: External research + Zendesk ticket for issues with no KB match.

2.4 Response Generation

- System enforces citation of ticket IDs in every solution.
- Retrieves up to 10 relevant chunks per query; combines for comprehensive answers.

- Escalates with `"ESCALATE_TO_HUMAN"` if no matching tickets are found or for edge/safety/warranty topics.

---

3. System Features

Feature	Description
Semantic search	Quick lookup through all historic support data
Cited answers	Every reply references source ticket IDs
Product/context awareness	Metadata enables nuanced, relevant responses
Conversation history	Session tracking with PostgreSQL chat log
Escalation workflow	Automated Zendesk ticket, Perplexity search
Quality controls	Enforced source citation, safety escalation

---

4. Testing and Results

- Dataset: 1000+ historic tickets, real Joolca support cases.
- Scenarios: Setup/troubleshooting/warranty and edge cases tested.
- Citations: 90% citation accuracy; responses cite correct tickets.
- Performance: Avg. response time ~11 seconds, down from manual 20+ seconds.
- Quality: 85% responses fully answered customer queries.

- Escalations: ~15% queries correctly escalated using fallback protocols.

Test Examples

- *“HOTTAP won’t ignite”*: Correct retrieval (Tickets J-0133/J-0225), step-by-step cited solution.
- *Warranty boundary question*: No matches; specialist handoff and customer notified.

---

5. Challenges & Solutions

Challenge	**Solution**
CSV parsing complexity	Built custom JS parser for quotes/multilines
Vector search latency	Implemented vector index, chunk batching
Maintaining context	Used overlapping/atomic chunking
API & rate limits	Batched embeddings, designed for failover
Workflow complexity	Modularized >35 nodes, added clear error handling
Data quality	Validated + cleaned source ticket/comment data

---

6. Limitations

- Only historical data: New tickets not auto-learned in current workflow.
- Chunk-level recall: Some complex tickets may require full thread synthesis.
- Manual maintenance: Updates needed for new product/KB content.
- English-only, text-only: No chat/image/modal/multilingual support yet.
- Basic analytics: No live dashboard for success rates or query analysis.

---

7. Future Improvements

Short-Term	**Long-Term**
Full ticket reconstruction (use ticket_id to fetch entire ticket narrative)	Image-based troubleshooting
Confidence scoring per answer	Predictive analytics/warranty modeling
Real-time ticket KB ingestion	Customer portal integration
Smart performance dashboard	Full personalization

---

8. Business Impact

Impact	**Detail**

Faster, consistent responses	Immediate, evidence-based answers
Knowledge retention	Historic agent expertise captured, reused
24/7 support	AI agent handles off-hour tickets
Cost effectiveness	~$12/month for 1000 queries, minimal maintenance

---

9. Deployment & Documentation

Deployment:

- Requires n8n (cloud/self-hosted), Supabase + vector extension, OpenAI & Zendesk API credentials.

Included Files:

- `final-rag.json` (all n8n workflow nodes)
- This report (PDF/Markdown)
- Embedded DB setup and code scripts

Documentation:
Submit as `/projects/joolca-support-rag/` (or your project repo), with this report and workflow JSON. Include any supplementary scripts or instructions as README in this folder. Add demo links if applicable.

---

10. Conclusion

Successfully delivered a fully functional, evidence-citing, and scalable RAG system for Joolca customer support. Project demonstrates the deployment of advanced workflow automation, semantic search, and LLM agents to solve real business support problems with traceability and human oversight.

Project Duration: July 2025 – September 2025
Submitted by: Ishaan Gosain