AI-POWERED DIGITAL CALL CENTER

Team: WeekendCoders

---

1. Problem Statement

AI-Powered Digital Call Center Using Autonomous AI Agents

Traditional call centers face significant challenges including high operational costs (average $25-35 per call), long customer wait times, inconsistent service quality, agent burnout, and limited 24/7 availability. Human agents spend 60-70% of their time on repetitive, routine inquiries that follow predictable patterns, while complex cases requiring human judgment often get delayed.

The challenge is to build an AI-powered digital call center that can autonomously handle customer interactions through voice and chat interfaces, while maintaining enterprise-grade safety, transparency, and knowing when to escalate to human agents.

---

2. Brief Explanation of How the Solution Works End to End

End-to-End Flow:

Customer → Voice/Chat Interface → AI Call Center → Resolution/Escalation

Step-by-Step Process:

1. **Customer Initiates Contact**

    - Customer starts a call via web interface (voice or chat)
    - System creates an interaction session and begins recording

2. **Speech-to-Text Processing (Voice)**

    - Browser Web Speech API converts speech to text in real-time
    - Continuous listening mode automatically captures customer speech
    - Audio waveform visualization provides feedback

3. **Primary Agent Analysis**

    - Detects customer **intent** (billing, order status, technical support, etc.)
    - Assesses customer **emotional state** (neutral, frustrated, satisfied, etc.)
    - Searches **Knowledge Base** for relevant solutions (semantic search with embeddings)
    - Generates context-aware response using LLM (Ollama/OpenAI/Gemini)
    - Reports **confidence score** (0.0-1.0)

4. **Supervisor Agent Review**

    - Reviews response quality and appropriateness
    - Checks compliance with policies
    - Validates tone matches customer emotion
    - Adjusts confidence if needed
    - Approves or flags for escalation

5. **Escalation Agent Decision**

    - Evaluates if human intervention is needed
    - Triggers based on: low confidence, emotional distress, explicit request, legal mentions
    - Creates support ticket with full context for human agents

6. **Response Delivery**

    - Text response delivered via chat
    - Text-to-Speech converts response to spoken audio (voice mode)
    - Quick reply suggestions displayed for common follow-ups

7. **Call Resolution**

    - Customer satisfaction detected from positive phrases
    - Call marked as "resolved" or "escalated"
    - AI-generated summary report available
    - Full conversation logged for audit and analytics

---

3. What is Unique or Innovative About Your Approach

Key Innovations:

Innovation	**Description**
Multi-Agent Architecture	Three specialized AI agents (Primary, Supervisor, Escalation) work in pipeline, each with distinct responsibilities. This mimics real call center hierarchy.
Confidence-Based Autonomy	AI operates autonomously only when confident (≥80%). Medium confidence triggers clarification questions. Low confidence escalates to humans.
Never-Override Safety Rules	Critical safety rules (legal mentions, threats, prohibited phrases) ALWAYS trigger escalation, regardless of LLM output. Deterministic safety layer.
Real Order Data Integration	AI accesses actual order database (orders.csv) to provide real tracking numbers, delivery dates, and status - not canned responses.
Semantic Knowledge Base Search	Uses sentence-transformer embeddings for true semantic search. "I want my money back" matches "refund policy" even without keyword overlap.
Vendor-Agnostic LLM Support	Supports OpenAI, Google Gemini, and local Ollama models. Switch providers without code changes. Enables privacy-first deployments.
Fast-Path Supervisor	High-confidence simple queries skip LLM-based supervisor review, reducing response time from 33s to 12s (63% improvement).
AI Transparency	Every decision includes reasoning steps. Complete audit trail. AI confirms it's an AI if asked.

Innovation	Description
Human Handoff with Context	When escalating, creates ticket with full conversation history, detected intent, emotion trajectory, and recommended actions for human agent.
Downloadable Call Reports	AI-generated summary reports with customer issue, resolution, topics, sentiment journey, and recommendations.

4. Target Users and Use Cases

4a. Primary Target Users / Customer Segment

User Type	Description
E-commerce Companies	Handle order inquiries, returns, refunds at scale
SaaS Providers	Technical support, billing questions, account management
Banks & Financial Services	Account inquiries, transaction disputes (with human escalation for sensitive issues)
Telecom Companies	Service inquiries, plan changes, technical troubleshooting
Healthcare Administration	Appointment scheduling, insurance inquiries (non-medical)
Enterprise IT Help Desks	Password resets, basic troubleshooting, ticket creation

4b. Main Use Cases

Use Case 1: Order Status Inquiry (80% of calls)

Customer: "Where is my order ORD10024?"

AI: Looks up order → "Your order is SHIPPED. Tracking: 1Z999AA10123456799. Estimated delivery: Feb 5th."

Result: Resolved autonomously in ~12 seconds

Use Case 2: Escalation to Human Agent

Customer: "I've been waiting 3 weeks for my refund! I want to speak to a manager NOW!"

AI: Detects frustration + escalation trigger → Creates ticket → Routes to human agent

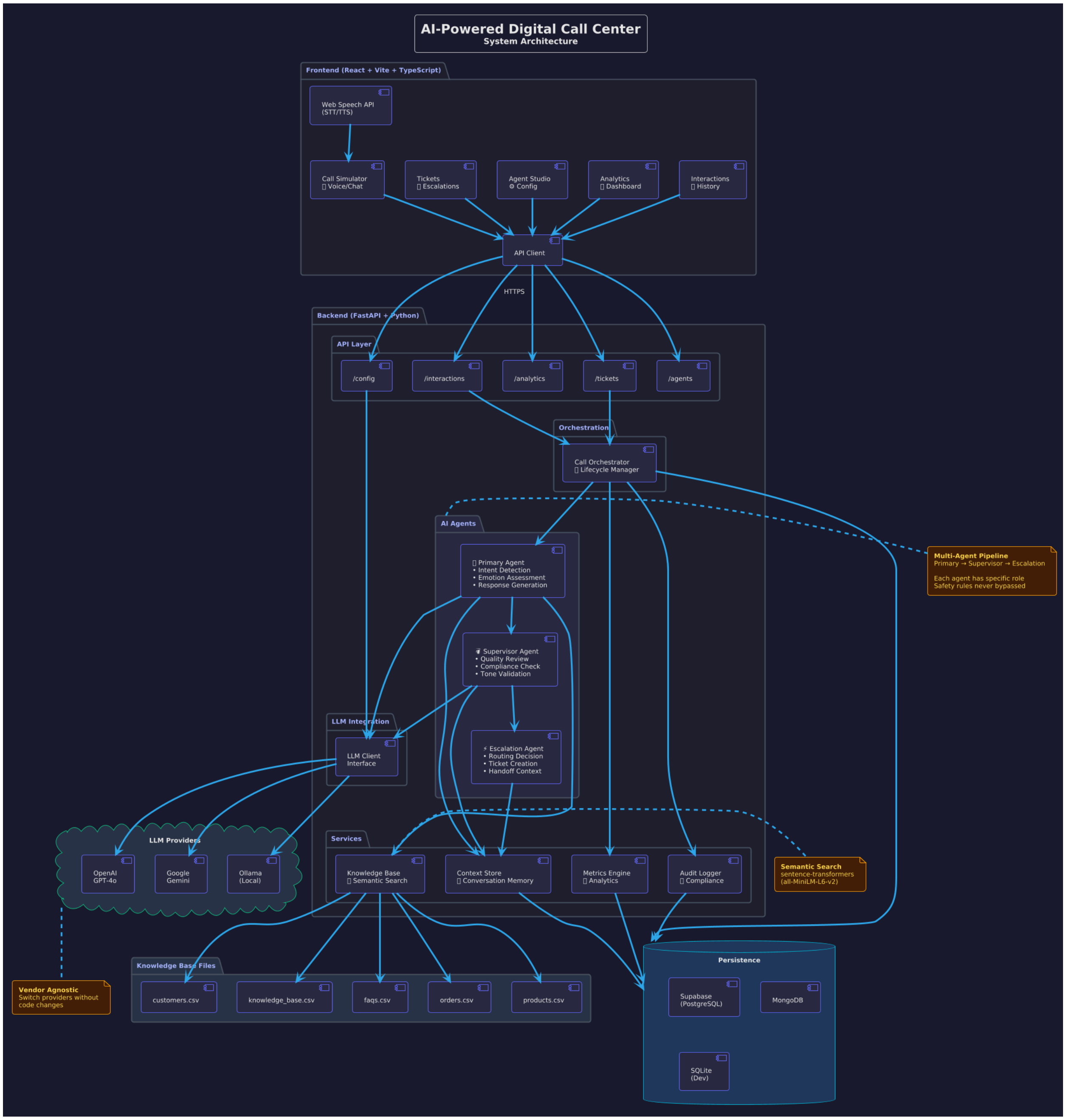
Result: Seamless handoff with full context

4c. Assumptions About User Environment

Aspect	**Assumption**
Device	Modern web browser (Chrome, Firefox, Safari, Edge) with microphone access
Connectivity	Stable internet connection (minimum 1 Mbps for voice)
Skills	No technical skills required; familiar with voice assistants or chat interfaces
Environment	Reasonably quiet for voice input; text chat works anywhere
Accessibility	Works on desktop and tablet; mobile-responsive design

5. Architecture and Technical Design

5a. High-Level Architecture


AI-Powered Digital Call Center
System Architecture
Frontend (React + Vite + TypeScript)
Web Speech API (STT/TTS)
Call Simulator ▯ Voice/Chat
Tickets ▯ Escalations
Agent Studio ⚙ Config
Analytics ▯ Dashboard
Interactions ▯ History
API Client
HTTPS
Backend (FastAPI + Python)
API Layer
/config
/interactions
/analytics
/tickets
/agents
Orchestration
Call Orchestrator ▯ Lifecycle Manager
AI Agents
▯ Primary Agent • Intent Detection • Emotion Assessment • Response Generation
🛡 Supervisor Agent • Quality Review • Compliance Check • Tone Validation
⚡ Escalation Agent • Routing Decision • Ticket Creation • Handoff Context
LLM Integration
LLM Client Interface
LLM Providers
OpenAI GPT-4o
Google Gemini
Ollama (Local)
Services
Knowledge Base ▯ Semantic Search
Context Store ▯ Conversation Memory
Metrics Engine ▯ Analytics
Audit Logger ▯ Compliance
Knowledge Base Files
customers.csv
knowledge_base.csv
faqs.csv
orders.csv
products.csv
Persistence
Supabase (PostgreSQL)
MongoDB
SQLite (Dev)
Multi-Agent Pipeline
Primary → Supervisor → Escalation
Each agent has specific role
Safety rules never bypassed
Semantic Search
sentence-transformers
(all-MiniLM-L6-v2)
Vendor Agnostic
Switch providers without code changes

5b. Technologies, Frameworks, Libraries, Models and Tools

Category	**Technologies Used**
Frontend Framework	React 18, Vite, TypeScript
UI Components	Custom CSS Modules, Lucide Icons
Voice (Browser)	Web Speech API (SpeechRecognition, SpeechSynthesis)
Backend Framework	FastAPI (Python 3.9+), Uvicorn (ASGI)
Data Validation	Pydantic v2
LLM Providers	OpenAI API, Google Gemini API, Ollama (local)
LLM Models	GPT-4o, GPT-4o-mini, Gemini 2.0 Flash, Llama 3.1 8B
Semantic Search	sentence-transformers (all-MiniLM-L6-v2)
Database	Supabase (PostgreSQL), MongoDB (optional), SQLite (dev)
Authentication	JWT (python-jose), Password hashing (passlib)
HTTP Client	httpx (async)
Environment Config	python-dotenv
Version Control	Git, GitHub

---

6. Implementation Details

6a. Current Implementation Status

Component	**Status**	**Details**
Core Agent Pipeline	✅ Complete	Primary → Supervisor → Escalation agents fully functional

Component	Status	Details
Knowledge Base Search	✅ Complete	Semantic search with embeddings, real order lookups
LLM Integration	✅ Complete	OpenAI, Gemini, Ollama all supported and tested
Voice Input/Output	✅ Complete	Browser-based STT/TTS with waveform visualization
Human Escalation	✅ Complete	Ticket creation, live chat sessions
Analytics Dashboard	✅ Complete	Metrics, trends, call statistics
Call Summary Reports	✅ Complete	AI-generated summaries, downloadable reports
Agent Studio	✅ Complete	Prompt editing, LLM configuration, testing
Authentication	✅ Complete	JWT-based login, demo user
Persistence	✅ Complete	Supabase/MongoDB/SQLite support

Overall Status: END-TO-END PROTOTYPE — All core features implemented and functional.

6b. Data Used

Data Type	Source	Processing
Knowledge Base	`knowledge_base.csv` (46 entries)	Agent procedures for billing, technical, orders, etc.
Orders	`orders.csv` (25 sample orders)	Order ID, status, tracking, delivery dates
Products	`products.csv` (21 products)	Product info, troubleshooting steps
Customers	`customers.csv` (21 customers)	Customer profiles, membership tiers

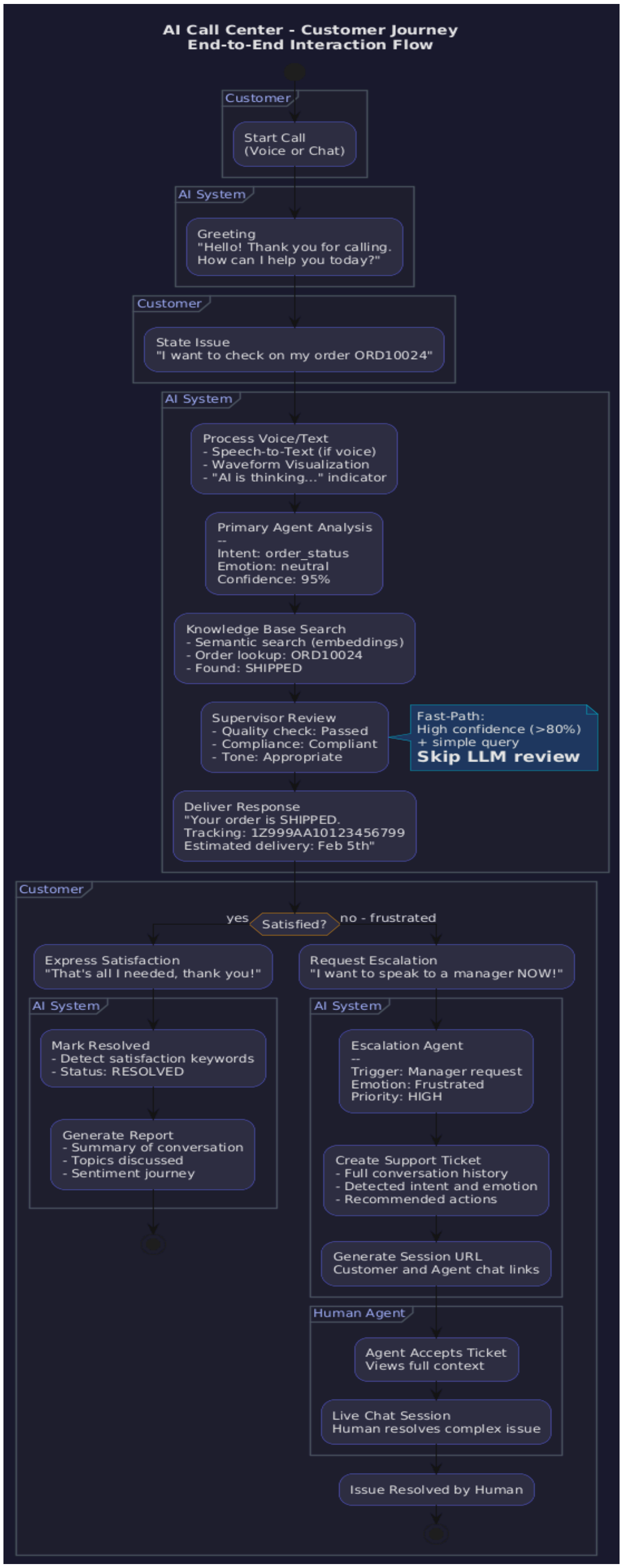
Data Type	Source	Processing
FAQs	`faqs.csv`	Frequently asked questions and answers

Data Processing:

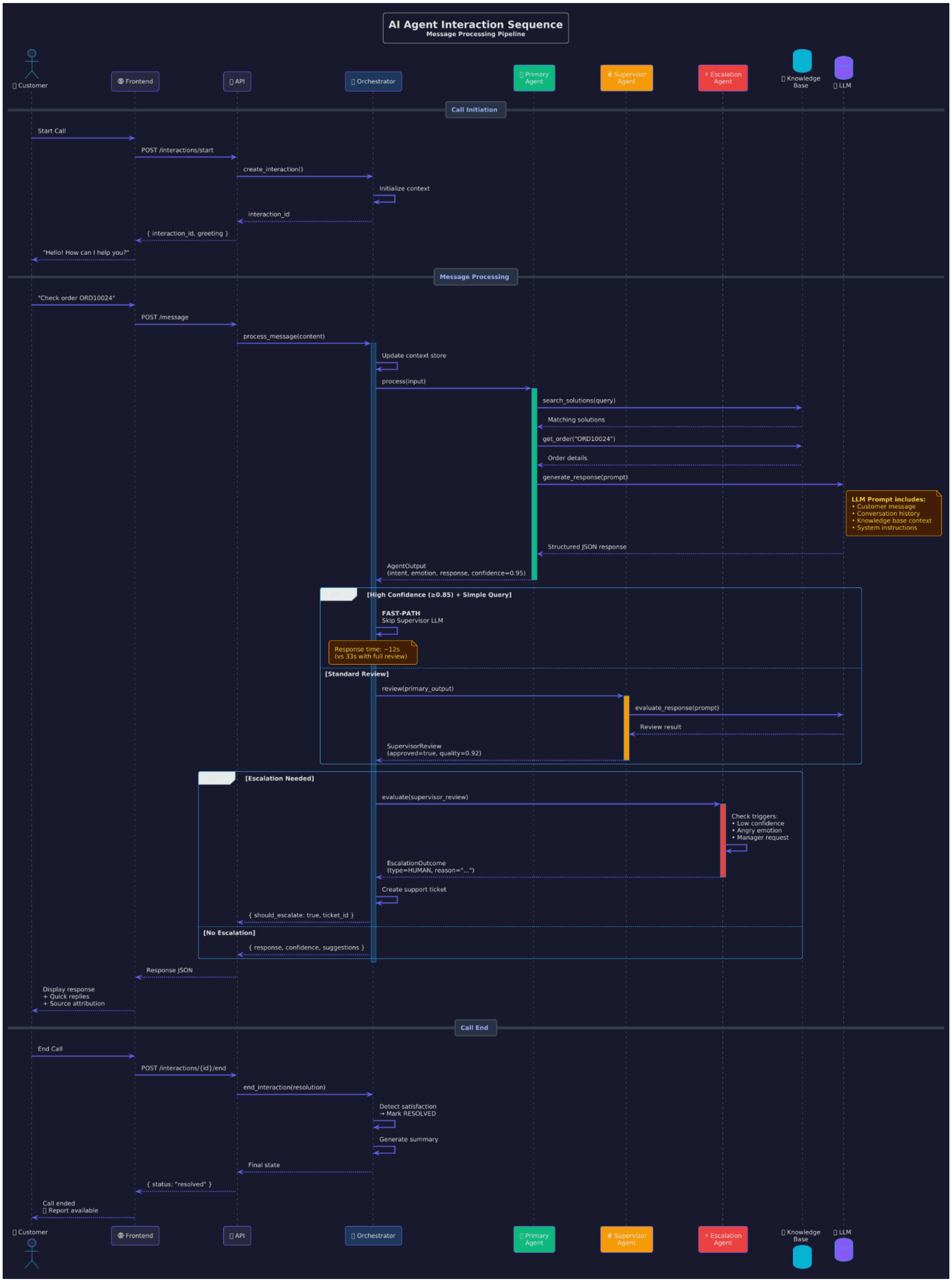
1. CSV files loaded into memory at startup
2. Text fields combined and embedded using sentence-transformers
3. Cosine similarity search for semantic matching
4. Direct lookups by ID for order/customer queries

7. User Experience and Workflow

Customer Journey Diagram



Sequence Diagram



Key UX Features

1. **Real-time Feedback** — Typing indicator, processing status, waveform visualization
2. **Quick Replies** — One-click suggested responses
3. **Source Attribution** — Shows where information came from
4. **Continuous Voice** — No need to press button for each utterance
5. **Call Summary** — AI-generated report available at end

---

8. Challenges and Limitations

8a. Technical Challenges Faced

Challenge	**Solution Implemented**
LLM Response Time	Implemented fast-path supervisor (skip LLM for high-confidence queries). Reduced 33s → 12s.
LLM Reading Internal Procedures	Updated prompts to explicitly label KB content as "internal guidelines". Added conversion layer.
Ollama Nested JSON Responses	Added `_normalize_string_list()` to handle dict responses in arrays.
Empty Orders Database	Populated with 25 sample orders for demo.
Call Classification as "Abandoned"	Added satisfaction detection to mark calls as "resolved".
Remote Ollama Connection	Updated client to support custom base URLs for LAN/remote servers.
Gemini API Compatibility	Used direct REST calls instead of Python SDK for better control.
Browser Speech API Limitations	Handled permission errors, added continuous recognition mode.

8b. Current Limitations

Limitation	Impact	Future Solution
No Phone (PSTN) Integration	Voice only works via browser	Integrate Twilio/Vonage for phone calls
Single Tenant	One organization per deployment	Add multi-tenancy with row-level security
In-Memory LLM Config	API keys lost on restart	Add encrypted key storage
No Real-Time WebSocket	Polling-based live sessions	Implement WebSocket for instant updates
English Only	No multi-language support	Add translation layer, multilingual models
No Billing Integration	Cannot check actual account balances	Integrate with billing APIs
Limited Product Catalog	21 sample products	Connect to real product database

---

9. Future Enhancements and Roadmap

Phase 1: Production Ready (1-2 months)

- ☐ Phone (PSTN) integration with Twilio
- ☐ Multi-tenancy with organization isolation
- ☐ Encrypted secret storage (HashiCorp Vault)
- ☐ WebSocket for real-time updates
- ☐ Redis session storage

Phase 2: Enterprise Features (2-4 months)

- ☐ Multi-language support (10+ languages)
- ☐ CRM integration (Salesforce, HubSpot)
- ☐ Advanced analytics and reporting
- ☐ Custom LLM fine-tuning per tenant
- ☐ Role-based access control
- ☐ SLA monitoring and alerts

Phase 3: Scale & Optimize (4-6 months)

- □ Kubernetes deployment
- □ Auto-scaling based on call volume
- □ A/B testing for response strategies
- □ Voice cloning for branded voices
- □ Sentiment-based routing
- □ Predictive escalation (flag issues before they escalate)

---

10. Team Member Details

Team Member 1

Field	**Details**
Name	Ruturaj Solanki
Company	Tata Consultancy Services
Phone Number	9512373608
Email	ruturaj.solanki@tcs.com

Team Member 2

Field	**Details**
Name	Sakshi Adarkar
Company	Tata Consultancy Services
Phone Number	7758076356
Email	sakshi.adarkar@tcs.com

Team Member 3

Field	Details
Name	Anamay Mishra
Company	Tata Consultancy Services
Phone Number	7905942713
Email	anamay.mishra@tcs.com

Team Member 4

Field	Details
Name	Kheya Das
Company	Tata Consultancy Services
Phone Number	9835451056
Email	kheya.das@tcs.com

Team Member 5

Field	Details
Name	Brajesh Singh Chouhan
Company	Tata Consultancy Services
Phone Number	9406827444
Email	brajesh.chouhan@tcs.com

Appendix: Quick Demo Commands

Start the Application

```
# Backend
cd ai-call-center/backend
python3 -m uvicorn app.main:app --host 0.0.0.0 --port 8000 --reload
# Frontend (new terminal)
cd ai-call-center/frontend
npm run dev
```

Configure Ollama (Local LLM)

```
# On LLM server
OLLAMA_HOST=0.0.0.0:11434 ollama serve
# Pull model
ollama pull llama3.1:8b
```

Test Order Lookup

```
curl -X POST "http://localhost:8000/api/interactions/start" \
  -H "Content-Type: application/json" \
  -d '{"customer_id": "demo", "channel": "chat"}'
# Use returned interaction_id
curl -X POST "http://localhost:8000/api/interactions/{id}/message" \
  -H "Content-Type: application/json" \
  -d '{"content": "What is the status of order ORD10024?"}'
```

Document Version: 1.0
Last Updated: January 18, 2026
GitHub Repository: https://github.com/ruturajsolanki/AI_Hackathon