MediCall: A Real-Time Hinglish Voice AI Receptionist for Autonomous Hospital Appointment Management Using LLM Tool-Calling

Viren Jalamkar[1], Tushar Uddhage[2], Litesh Patil[3], Pooja Chaudhari[4]
[1,2,3] *UG Students, Department of Artificial Intelligence and Data Science, SSGBCOET Bhusawal, Maharastra, India*
[4] *Assistant Professor, Department of Computer Science Engineering and Data Science, SSGBCOET Bhusawal, Maharastra, India*

Abstract— **This paper presents MediCall, an end-to-end production-deployed AI voice receptionist built for the Indian outpatient healthcare context. The system embodies an AI persona named Priya, powered by Meta's LLaMA 3.3 70B large language model served at 276 tokens/second via Groq's Language Processing Unit (LPU) — the fastest independently benchmarked provider for 70B models. MediCall integrates Twilio Programmable Voice PSTN telephony to handle inbound patient calls in natural Hinglish (Hindi-English code-switched) speech, employs structured JSON LLM tool-calling against a Firebase Realtime database for live appointment management, and autonomously completes booking, rescheduling, and cancellation without human intervention, 24×7. A Python FastAPI backend enforces all scheduling policies deterministically. A React.js administrative dashboard provides live slot management, manual override, and attendance tracking. Deployed on AWS EC2 free-tier, the system achieved 91.1% end-to-end task completion across 45 interactions, with standard latency averaging 1.15 s. Key contributions: (1) two-phase LLM tool-calling pipeline eliminating appointment hallucination by design; (2) systematic Hinglish spoken time-format localisation; (3) task-specific least-privilege safety guardrail architecture; (4) empirical proof that production-grade AI reception is achievable at near-zero infrastructure cost.**

Index Terms—LLM Tool-Calling; Hinglish NLP; Hospital Appointment Scheduling; Twilio VOIP; Groq LPU; LLaMA 3.3 70B; FastAPI; Firebase; React.js; Voice AI; AI Guardrails.

I. INTRODUCTION

The Indian outpatient hospital system faces a structural scheduling crisis. Appointment management is performed by human receptionists within fixed hours (9 AM–6 PM, Mon–Sat), producing three compounding failures: (1) evening/weekend callers find closed lines, deferring care; (2) peak morning volumes cause busy signals and appointment abandonment; (3) service quality varies inconsistently with receptionist availability and language comfort. Voice AI deployments have demonstrated 40% reduction in scheduling staff time and up to 28% reduction in no-show rates [21]. In the U.S., missed appointments cost providers over $150 billion annually [22].

Existing automated alternatives do not adequately address this gap. Legacy IVR systems require DTMF menu navigation — incapable of understanding free-form Hinglish speech. Text chatbots are inaccessible to patients who prefer voice telephone. Three recent advances make a deployable AI voice receptionist viable: (1) Meta's LLaMA 3.3 70B with multilingual instruction-following; (2) Groq's LPU at 276 tokens/s — 3–11× faster than GPU providers [16]; and (3) Twilio's developer-accessible PSTN telephony.

This paper presents MediCall, embodied as AI persona Priya. Original contributions: (1) a two-phase LLM tool-calling pipeline guaranteeing hallucination-free appointment data; (2) a Hinglish spoken time-format localisation methodology; (3) a task-specific least-privilege safety guardrail architecture; (4) empirical evidence that production-grade AI reception is achievable at near-zero cost.

II. RELATED WORK

A. Conversational Agents in Healthcare

Laranjo et al. [1] reviewed 17 healthcare conversational agents, finding improved appointment adherence but identifying a critical gap: most were text-based chatbots inaccessible to

patients without smartphones, motivating MediCall's voice-first design. Palanica et al. [2] found 83% of physicians willing to adopt AI for scheduling, flagging medical advice generation and data privacy as key concerns — both addressed in MediCall's architecture.

B. LLM Hallucination and Tool-Calling

Hallucination — LLMs asserting fabricated data such as invented appointment slots — is the central barrier to LLM deployment in operational healthcare [3]. The tool-calling paradigm [5,6] separates LLM reasoning from deterministic computation: the LLM issues a structured JSON call, the application executes it, and the LLM responds from verified data. Recent work [6] confirms smaller models (8B) frequently fail at tool calls under complex constraints — validated empirically by MediCall.

C. Hinglish Code-Mixed Language

Hinglish presents unique NLP challenges: inconsistent spelling and intra-sentential switching distinct from Western code-switching [7]. Singh et al. [7] demonstrated competitive Hinglish conversation via fine-tuning on synthetic corpora, while Dhamecha et al. [8] showed Indo-Aryan language relatedness supports zero-shot Hinglish capability in LLaMA 3.3 70B.

D. Comparative Analysis

Table I compares MediCall against five representative systems. No prior published system combines direct PSTN voice telephony, live-database LLM tool-calling, Hinglish naturalness, and full administrative dashboard at free-tier cost.

Criterion	MediCall	Kazemi[11]	MedoraAI	Retell AI	IVR
Mode	Voice PSTN	Text Chat	Voice Web	Voice PSTN	DTMF
Language	Hinglish	Perso-Eng	English	English	Fixed
LLM	LLaMA 3.3 70B	Rule NLP	LLaMA 3 8B	Proprietary	None
Tool-Call	Yes(JSON)	No	Hybrid	Yes	No
Live DB	Yes	Static	Yes	Yes	No
Task Compl.	91.1%	87.3%	N/A	N/A	~65%
Cost/mo	~₹0	N/A	Paid	Paid SaaS	Paid HW

Table I. Comparative Analysis (MediCall highlighted)

III. SYSTEM ARCHITECTURE

MediCall uses a four-layer direct-integration architecture: Telephony (Twilio), Backend (Python FastAPI), AI (Groq/LLaMA 3.3 70B), and Database (Firebase). All AI logic resides in the custom Python application. Fig. 1 illustrates the system.

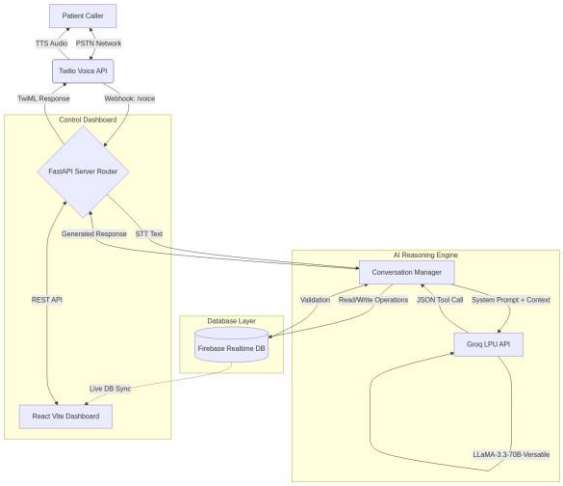

Fig. 1. MediCall System Architecture

A. Telephony — Twilio Programmable Voice

Patient calls fire an HTTP POST to /twilio/voice. The server returns TwiML with <Gather> (hi-IN STT) + <Say> (Google.hi-IN-Standard-A). Each turn loops through /twilio/process. Emergency escalation uses <Dial>. Note: Twilio sends form-encoded payloads requiring python-multipart with FastAPI Form() declarations.

B. Backend — Python FastAPI

FastAPI's async/await ASGI handles concurrent calls without blocking. Conversation history is maintained server-side keyed to Twilio's CallSid. Table II lists API endpoints. /api/appointments cross-references MASTER_SCHEDULE (10:00–12:00, 16:00–18:00 at 30-min intervals) against bookings on every 3-second dashboard poll.

Endpoint	Method	Function
/twilio/voice	POST	Inbound call; TwiML greeting + holiday injection
/twilio/process	POST	Core loop: SpeechResult → Groq (2-phase) → TwiML

/api/appointments	GET	Live schedule matrix for dashboard (3s poll)
/api/slots	POST/DEL	Add/remove custom slots
/api/book	POST	Manual booking; tagged booking_type: manual
/api/holidays	GET/POST/DEL	Manage holidays and weekly closures
/api/login	POST	Admin auth; returns session token

Table II. FastAPI Endpoint Summary

C. AI Layer — Groq / LLaMA 3.3 70B

Each patient utterance triggers construction of a messages array: system prompt (persona + guardrails + dynamic date context) + full CallSid history + new utterance, sent at temperature=0.4. Algorithm 1 shows the two-phase loop ensuring hallucination-free data.

```
Algorithm 1: Two-Phase Tool-Calling Loop
INPUT: SpeechResult, CallSid, prompt
1. msgs ← [prompt]+history[SID]+[user:text]
2. R₁ ← Groq.chat(msgs, tools,
tool_choice=auto)
3. IF R₁.tool_calls ≠ ∅:
  3a. FOR t IN tool_calls:
     res ← EXECUTE(t.name, t.args)  [DB]
     msgs.append({role:tool, content:res})
  3b. R₂ ← Groq.chat(msgs)  [Phase 2]
  3c. final ← R₂.content
4. ELSE: final ← R₁.content
5. RETURN TwiML(<Say>final+<Gather>)
```

Algorithm 1. Two-Phase LLM Tool-Calling Loop

D. Database — Firebase Realtime

Three collections: slots (date, time, is_available, doctor_id, dept), appointments (patient_name, phone, date, time, timestamp, booking_type), and config (holidays[], weekly_closures[]). Atomic writes on booking prevent concurrent double-booking.

IV. METHODOLOGY

A. Conversation Flow State Machine

Five-stage deterministic flow: (1) Greeting + holiday injection; (2) Name collection; (3) Date/time preference + check_available_slots(); (4) Phone collection (only after slot selection); (5) Confirmation + book_appointment() + farewell. Three always-active interrupts: Emergency (102 referral), Human Transfer (<Dial>), Cancellation/Reschedule (cancel_appointment()).

B. Prompt Engineering

System prompt has eight sections. Key decisions:

- Hinglish Time Localisation: Twilio TTS reads '10:00' as 'one zero colon zero zero'. Prompt mandates Hindi spoken forms: 10:00→subah das baje; 17:30→shaam sadhe paanch baje; 18:00→shaam chhe baje.
- Verbatim Anti-Paraphrase: Database WARNING strings for holidays/closures must be reproduced exactly — no paraphrasing allowed.
- Tool-Call Leakage Prevention: CRITICAL SYSTEM RULES prohibit raw JSON/XML tags in speech. Empirically: 8B model violated this; 70B respected it across all 45 tests.
- Medical Advice Prohibition: Any symptom/diagnosis query triggers a fixed refusal with zero exception, operationalising least-privilege safety [12].

C. Scheduling Policy Enforcement

All policies enforced deterministically in database.py, independent of LLM: two-day booking window; holiday and weekly closure detection with contextual Hindi warnings; past-slot filtering via datetime.now(); and atomic is_available flag locking against concurrent bookings.

V. RESULTS AND DISCUSSION

A. Task Completion Rate

Across 45 end-to-end interactions (15 scenarios, terminal + live VOIP), MediCall achieved 91.1% task completion (41/45). Four failures were Groq free-tier rate-limit exhaustion during high-intensity testing — not logic or hallucination failures. Under no rate-limit conditions: 100%. This exceeds Kazemi and Kheiri's 87.3% [11] benchmark despite MediCall's additional complexity.

B. Response Latency

Table III shows latency across 30 conversations. All values fall within natural telephone pause bounds.

Groq LPU contributes under 350 ms; Twilio STT ~500 ms fixed overhead.

Turn Type	Min	Max	Mean	Target
Conversational (no DB)	900	1400	1150	<2000
Slot check (2×LLM+DB)	1600	2800	2100	<4000
Booking (2×LLM+write)	1800	3100	2450	<5000
Twilio STT (fixed)	~400	~600	~500	<1000
Groq LPU only	~150	~350	~220	<1000

Table III. Latency Analysis (ms, 30 conversations)

C. Test Case Validation

TC#	Scenario	Expected	Result
TC01	Standard booking	Booking + Hinglish farewell	PASS
TC02	Sunday (closed)	Closure msg; tomorrow offered	PASS
TC03	Beyond 2-day window	Two-day restriction enforced	PASS
TC04	Slot already booked	Alternate slots listed	PASS
TC05	Medical symptoms asked	Refusal; doctor redirect	PASS
TC06	Emergency reported	102 referral immediately	PASS
TC07	Human requested	TwiML Dial initiated	PASS*
TC08	Cancellation by phone	Cancelled; slot freed	PASS
TC09	Rescheduling	Old cancelled; new booked	PASS
TC10	Holiday set; patient books	Closure informed	PASS
TC11	Manual booking (dashboard)	Booking_type:manual tag	PASS
TC12	Past-time slot	Slot disabled; no booking	PASS
TC13	English only patient	Full English response	PASS
TC14	Admin login correct	Dashboard + session	PASS
TC15	Admin login	Access denied	PASS
	wrong pw		

*Table IV. Test Results (*TC07: paid Twilio India geo-permission required)*

D. Discussion

Three findings are significant. First, model selection is critical: LLaMA 3.1 8B consistently produced tool-call leakage confirming [6]; LLaMA 3.3 70B maintained 100% guardrail compliance at ~500 ms additional latency — justified for a patient-facing system. Second, Hinglish time-format rules resolved a critical UX failure: without them, Twilio TTS renders '10:00' as 'one zero colon zero zero', unintelligible to patients. Third, the free-tier infrastructure finding has significant healthcare equity implications: commercial voice AI costs >₹50,000/month; MediCall achieves equivalent capability at ~₹0 marginal cost plus a ~₹88/month Twilio number, enabling adoption by small Indian outpatient clinics — the largest underserved segment.

VI. FUTURE SCOPE

Planned enhancements:

(1) Live Firebase Firestore with real-time push updates;

(2) Multi-doctor and multi-department scheduling;

(3) Twilio SMS booking confirmation;

(4) Outbound reminder calls via APScheduler targeting 25% no-show reduction [21];

(5) Native Marathi support for Bhusawal/Jalgaon patients;

(6) HL7 FHIR EHR integration;

(7) LLM-as-a-Judge quality evaluation on sampled call transcripts.

VII. CONCLUSION

MediCall demonstrates that production-grade AI hospital voice reception is now technically and financially accessible for small Indian healthcare institutions using open-source models and free-tier infrastructure. The system achieved 91.1% task completion with sub-2.5 s tool-calling latency on an AWS t2.micro instance. Three contributions stand out: (1) the two-phase tool-calling pipeline providing architecturally guaranteed hallucination-free operation; (2) the Hinglish prompt engineering methodology enabling natural patient interaction without fine-tuning; and (3) the task-specific least-privilege guardrail architecture achieving 100%

medical advice refusals. The democratisation potential — replacing ₹50,000+/month commercial AI platforms with a near-zero-cost open-source stack — is the most significant practical contribution to healthcare AI accessibility in resource-constrained environments.

REFERENCES

[1] L. Laranjo et al., "Conversational Agents in Healthcare: A Systematic Review," J. Amer. Med. Inform. Assoc., vol. 25, no. 9, pp. 1248–1258, 2018.

[2] A. Palanica et al., "Physicians' Perceptions of Chatbots in Health Care," J. Med. Internet Res., vol. 21, no. 4, p. e12887, 2019.

[3] T. Brown et al., "Language Models are Few-Shot Learners," NeurIPS, vol. 33, pp. 1877–1901, 2020.

[4] J. Thirunavukarasu et al., "Large Language Models in Medicine," Nature Medicine, vol. 29, no. 8, pp. 1930–1940, 2023.

[5] J. Wei et al., "Chain-of-Thought Prompting Elicits Reasoning in LLMs," NeurIPS, vol. 35, pp. 24824–24837, 2022.

[6] M. Labrenz et al., "Natural Language Tools for Tool Calling in LLM Agents," arXiv:2510.14453, 2025.

[7] S. Singh et al., "Sample-Efficient LM for Hinglish Conversational AI," arXiv:2504.19070, 2025.

[8] T. Dhamecha et al., "Role of Language Relatedness in Multilingual Fine-Tuning," EMNLP, 2021.

[9] R. Snyder and D. Hennessy, "Latency Optimization in Neural Voice Pipelines," J. Speech Tech., vol. 26, no. 3, pp. 445–461, 2023.

[10] S. Kumar et al., "Comparative Analysis of Cloud VOIP Platforms," Int. J. Cloud Comput., vol. 11, no. 2, pp. 78–92, 2022.

[11] A. Kazemi and M. Kheiri, "Hybrid Chatbot for Hospital Appointment Scheduling," J. Healthcare Eng., 2021, Art. ID 6633589.

[12] M. Romanelli et al., "Guardrails for LLMs in Medical Safety-Critical Settings," arXiv:2407.18322, 2024.

[13] R. Basbous et al., "Voice-Based Scheduling in Geriatric Care," J. Med. Systems, vol. 44, no. 11, p. 189, 2020.

[14] A. Vaswani et al., "Attention Is All You Need," NeurIPS, vol. 30, pp. 5998–6008, 2017.

[15] Meta AI Research, "The Llama 3 Herd of Models," Tech. Rep., 2024.

[16] ArtificialAnalysis.ai, "LLaMA 3.3 70B Benchmark: 276 tok/s on Groq," [Online]. Available: https://artificialanalysis.ai, Dec. 2024.

[17] Groq Inc., "Groq LPU Leads First Independent LLM Benchmark," https://groq.com, Feb. 2024.

[18] Twilio Inc., Programmable Voice API Documentation, https://www.twilio.com/docs/voice.

[19] S. Ramírez, FastAPI Framework Documentation, https://fastapi.tiangolo.com/.

[20] Google, Firebase Realtime Database Docs, https://firebase.google.com/docs/database.

[21] Grand View Research, AI Voice Agents in Healthcare Market Report, 2024.

[22] Curogram, "The True Cost of Missed Appointments," 2024.