Quickup Call Agent

An AI voice-based phone-support system for food delivery

Local Whisper · STT | Local Llama 3.2 · LLM | ElevenLabs · TTS | MySQL

Project documentation · Generated for the Quickup demo

Contents

1. What it is

A simulated AI phone-call support system for **Quickup**, a food-delivery service. A rider with an active order calls support; the AI agent identifies them, understands their problem, calls the restaurant on the rider's behalf, gets a decision, then relays it back to the rider — with both legs of the call voiced naturally.

It is a real working voice-agent system — not a video, not slides — using mostly free, locally hosted components.

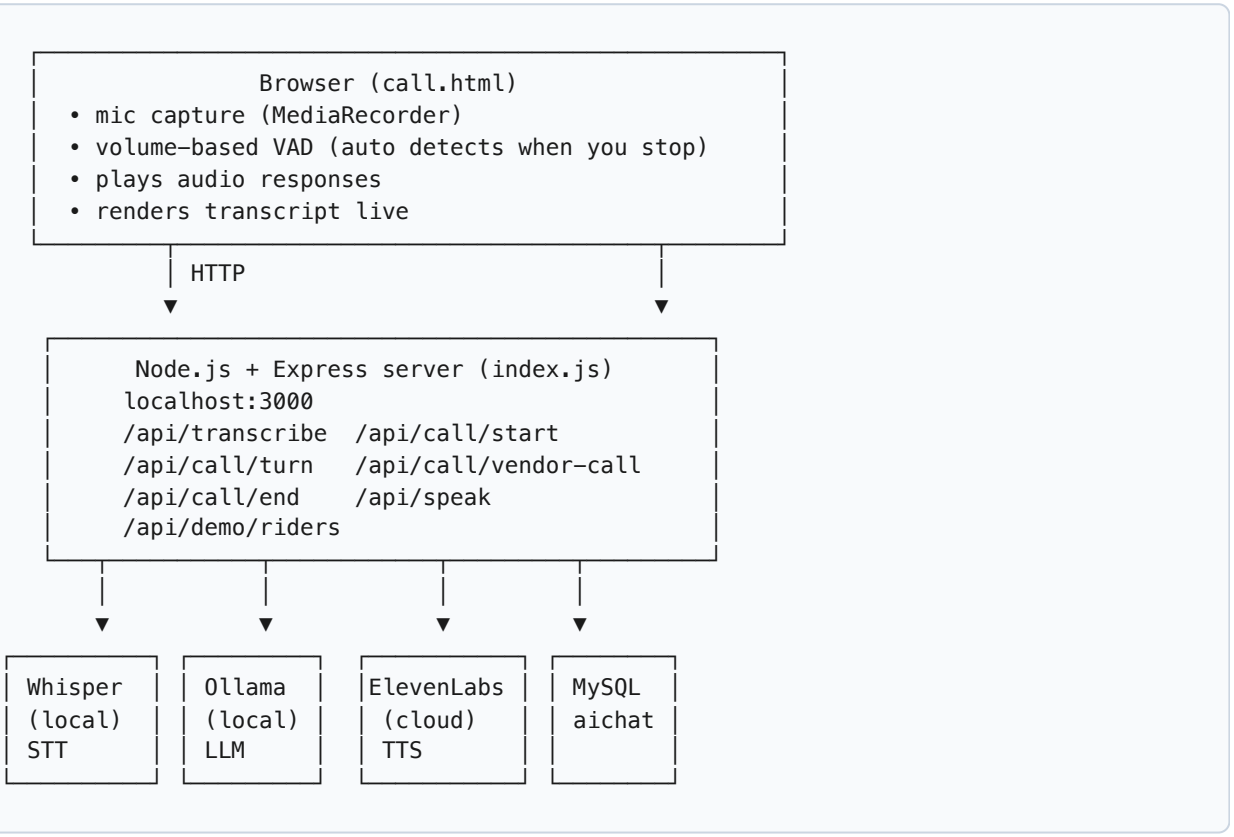
2. The full demo scenario

1. Rider Ahmad has already picked up order `ORD-5821` (1× Large Pepperoni Pizza, 2× Pepsi) from **Pizza Hut Gulberg** for customer **Sara Khan**.
2. Ahmad arrives at Sara's address (House 5, DHA Phase 3) — she is not at the door, not answering her phone.
3. Ahmad calls Quickup support and gets through to AI agent **Sarah**.
4. Sarah identifies Ahmad and his active order instantly (from the database).
5. Ahmad explains the problem.
6. Sarah says *"Let me call Pizza Hut now, please hold."* and the demo auto-transitions.

7. Sarah (different ElevenLabs voice for the restaurant) speaks to **George**, the Pizza Hut manager.
8. George decides to cancel the order, refund Sara, and have the rider dispose of the parcel.
9. The demo transitions back to the rider call.
10. Sarah relays the decision to Ahmad. Call ends. Transcript is saved to MySQL.

Total demo length: about 90 seconds.

3. Architecture



4. Tech stack

Backend

- **Node.js + Express 5** — HTTP server, all API endpoints
- **MySQL 9.0+** — relational data and 768-d vector embeddings (originally for the older RAG features)
- **multer** — file uploads (audio + PDF)
- **bcryptjs + jsonwebtoken** — auth (left over from older parts; the call demo is unauthenticated)
- **dotenv** — config loading

AI / Voice services

Component	What runs it	Cost	Where
Speech-to-text	Whisper.cpp + base.en model, Metal-accelerated	Free	localhost (your Mac)
Conversation LLM	Ollama + Llama 3.2 3B	Free	localhost:11434
Text-to-speech	ElevenLabs	Free tier (~10k chars/mo)	cloud API
Audio conversion	ffmpeg	Free	brew install

Frontend

- Vanilla HTML/CSS/JS (no framework)
- Web Audio API for microphone capture
- MediaRecorder API for audio recording
- AnalyserNode for volume-based voice activity detection (VAD)

5. File map

```
AiProject/
├── .env                                ← all config (API keys, model names, DB creds)
├── index.js                            ← Express server, all endpoints
├── views/
│   ├── call.html                       ← demo UI: lobby → ringing → in-call → vendor-cal
│   ├── call.js                         ← orchestration, mic, VAD, playback
│   ├── chat.html / chat.js             ← older chat UI (separate from call demo)
│   └── dashboard.html / studio.html    ← original RAG dashboard
├── services/
│   ├── chunker.js                      ← semantic text chunking (for old RAG)
│   └── embeddings.js                   ← Gemini 768-d embeddings + cosine similarity
├── migrations/
│   ├── 001_add_vector_support.sql      ← original RAG tables
│   └── 002_quickup_demo_data.sql       ← Quickup tables + seed data
├── scripts/
│   ├── backfill_embeddings.js          ← bulk-embed old knowledge
│   └── setup_whisper.js                ← downloads Whisper model + builds whisper.cpp
└── uploads/                            ← multer staging (auto-cleaned per request)
```

6. API endpoints (call demo)

Method + Path	Purpose	Returns
`GET /call.html`	Serve the demo UI	HTML
`GET /api/demo/riders`	List the 3 demo riders + their active orders	JSON array
`POST /api/call/start`	Begin a call. Body `{rider_phone}`. Builds system prompt, returns greeting	`{call_id, greeting, rider, order, restaurant}`
`POST /api/transcribe`	Audio → text via local Whisper. Filters known noise hallucinations	`{text}`
`POST /api/call/turn`	One conversation turn. Detects `[CALL_VENDOR]` marker	`{text, action}`
`POST /api/call/vendor-call`	Generate a 4-line agent↔restaurant exchange in JSON	`{exchange[], resolution, summary}`
`POST /api/speak`	Text → MP3 via ElevenLabs. Body `{text, voice}`	`audio/mpeg`
`POST /api/call/end`	Persist transcript, free in-memory session	`{success, transcript}`

7. Data model

```
riders          (id, rider_code, name, phone, vehicle, rating)
restaurants     (id, name, branch, phone, address)
customers       (id, name, phone, address)
orders          (id, order_code, rider_id, restaurant_id, customer_id,
                 items, total_amount, status, notes)
call_sessions   (id, rider_id, order_id, started_at, ended_at, outcome, transcript)
```

Seed data (3 of each)

- **R-101 Ahmad Khan** → ORD-5821 @ Pizza Hut Gulberg (delivering, customer not present) ← the demo scenario
- **R-102 Bilal Hussain** → ORD-5822 @ KFC DHA (preparing)
- **R-103 Sara Iqbal** → ORD-5823 @ Burger King Johar Town (delivering)

8. How a single turn flows internally

```
[ User speaks into mic ]
         |  MediaRecorder webm/opus blob
         ▼
[ Browser VAD detects 1.2s of silence ]
         |  POST audio to /api/transcribe
         ▼
[ Server: ffmpeg → 16kHz mono WAV ]
         ▼
[ whisper-cli -m base.en → text file ]
         ▼
[ Filter Whisper hallucinations (silence noise) ]
         |  JSON {text}
         ▼
[ Browser appends "Rider: ..." to transcript ]
         |  POST {call_id, user_text} to /api/call/turn
         ▼
[ Server: chatLLM() → Ollama Llama 3.2 3B ]
         |  system prompt + full call history + new user message
         ▼
[ Detect [CALL_VENDOR] marker, strip it, set action ]
         |  JSON {text, action}
         ▼
[ Browser appends "Agent: ..." ]
         |  POST {text, voice:"agent"} to /api/speak
         ▼
[ Server: ElevenLabs API → MP3 ]
         |  audio/mpeg
         ▼
[ Browser plays via Audio, waits for it to finish ]
         ▼
[ if action === "call_vendor": runVendorCall() ]
[ else: startListening() again ]
         ▼
[ Loop ]
```

The vendor-call sub-flow does the same loop but with two voices and a single LLM call returning all 4 lines in one shot.

9. The Quickup agent's prompt

The agent system prompt tells the LLM:

- It is the Quickup support agent, on a phone call, polite, concise, English only
- Hard limit: replies under ~40 words, no markdown, no bullet lists
- Full caller profile: rider name/code/vehicle/rating + active order + restaurant + customer
- The likely problem (rider has parcel, customer absent)
- Step-by-step rules: greet → acknowledge → say "let me call them" → emit `[CALL_VENDOR]` → wait → relay decision
- Hard rule: never invent info outside the profile

The vendor-call writer prompt is single-shot — it generates the entire 4-line exchange + resolution as JSON in one model call.

10. Configuration (.env)

```
# Gemini (cloud LLM, alt provider)
GEMINI_API_KEY=...
GEMINI_CHAT_MODEL=gemini-2.5-flash-lite

# Which LLM the agent uses
LLM_PROVIDER=ollama
OLLAMA_URL=http://localhost:11434
OLLAMA_MODEL=llama3.2:3b

# ElevenLabs (TTS)
ELEVENLABS_API_KEY=sk_...
ELEVENLABS_VOICE_ID=EXAVITQu4vr4xnSDxMaL          # Sarah — agent voice
ELEVENLABS_VENDOR_VOICE_ID=JBFqnCBsd6RMkjVDRZzb   # George — restaurant voice

# Auth + DB
JWT_SECRET=...
DB_HOST=localhost
DB_USER=root
DB_PASSWORD=
DB_NAME=aichat
```

11. What runs where

Component	Auto-start	Manual command if needed
MySQL	Yes	`brew services start mysql`
Ollama	Yes (LaunchAgent registered)	`brew services start ollama`
The Node server	No	`node index.js`

Your only demo command:

```
node index.js
```

12. Voice activity detection (VAD)

Defined in `views/call.js`:

```
const VAD = {
  silenceMs: 1200,            // wait this long after you stop talking
  speechRmsThreshold: 0.045,  // volume threshold to count as speech
  minSpeechMs: 700,           // ignore blips shorter than this
  maxRecordMs: 15000,         // hard cap per turn
};
```

It samples the mic's RMS volume every 80 ms. If volume goes above the threshold for at least 700 ms, it counts as speech. After speech, if 1200 ms of silence passes, the recording stops and is sent to Whisper.

13. Cost summary

Component	Setup cost	Per-demo cost
Whisper (STT)	Free	$0
Ollama + Llama 3.2 3B (chat)	Free	$0
ElevenLabs (TTS)	Free signup	~10 minutes of audio per month free
MySQL	Free	$0
ffmpeg, cmake	Free	$0

Total for a demo: $0. The ElevenLabs free tier easily covers many demo runs.

14. Limitations

1. **Not a real phone number** — works through a browser at localhost. Cloudflare Tunnel can give you a public URL for free, or Vapi/Twilio for an actual dial-in number.
2. **In-memory call sessions** — the `callSessions` Map dies if the server restarts mid-call. Fine for demos.
3. **The order is not actually cancelled in the DB** — the call demo logs the conversation but does not run an `UPDATE`. Easy to add.
4. **Single turn at a time** — no streaming. The user has to fully stop talking before the AI starts.
5. **English only** — Whisper model is base.en. Other languages need a different model.
6. **No interruption handling** — if you start talking while the AI is speaking, it does not stop. Production voice agents have barge-in detection.
7. **VAD is volume-based** — works but sensitive to background noise. Silero VAD would be much better.

15. Talking points for the Quickup demo

1. **It is modular.** Quickup can swap any layer. Want Azure Speech instead of Whisper? Change one endpoint. Want OpenAI instead of Llama? Change `LLM_PROVIDER`.
2. **It works offline.** STT and LLM run on the laptop. No internet required for the conversation logic.
3. **Cost at scale** — at production volume (e.g. 1,000 calls/day), the entire stack with ElevenLabs paid tier runs around $50–100/month versus $1k+/month with cloud LLM/STT services.
4. **Real database.** Every call persists rider, order, transcript, and outcome. Dashboards can be built immediately.
5. **The agent is RAG-able.** The existing `knowledge_data` table (with vector embeddings) lets each persona be trained on Quickup policies, FAQs, terms, etc. The current agent uses only the order profile; expanding to full knowledge retrieval is a few hours of work.

16. Future enhancements

Feature	Effort	Impact
Update order status to cancelled on call end	30 min	Makes the demo data real
Cloudflare Tunnel public URL	10 min	Anyone in the room can join
Real phone number via Vapi free credits	2 hours	Actual dial-in demo
Full RAG: agent uses Quickup policy docs	1 day	Agent answers policy/FAQ-type questions
Streaming TTS (talk while generating)	1 day	Sub-second perceived latency
Silero VAD instead of volume threshold	2 hours	Robust in noisy rooms