

VAIBHAV KOLEKAR

Full-Stack Developer

+91 78922 25626 va18havrk3@gmail.com Bengaluru, Karnataka



SUMMARY

Full-stack developer building real-time systems and distributed backend infrastructure. Shipped a voice-to-voice AI interview platform with a custom WebSocket-Gemini bridge and Redis-backed session recovery, and built a distributed job scheduler with atomic concurrency control and dead-letter handling — now powering a live API monitoring product. Comfortable owning a build end to end, from schema design to production deployment on AWS, Vercel, and Cloudflare.

PROJECTS

Interviu – AI Mock Interview Platform

interviu.pro · Code: github.com/va18hav/interviuPro

TypeScript · React.js · Node.js · WebSockets · Gemini Flash Live API · PostgreSQL · Redis · AWS S3 · Docker · AWS ECS

- Built a WebSocket server bridging client and Gemini Flash Live API connections for native voice-to-voice communication, handling bidirectional audio streaming throughout each interview session — **latency: 400-500ms** (observed across multiple interview sessions).
- Implemented Redis-backed session recovery — persisting transcript, system prompt, and Gemini resume handle per session — combined with exponential backoff retry logic, so a dropped connection mid-interview resumes instead of losing progress.
- Integrated the Claude Sonnet 4.6 API to generate structured, context-aware feedback from full interview transcripts, tailored to each session's interview context.
- Implemented JWT-based authentication from scratch — token issuance, validation middleware, and protected routes — rather than relying on a third-party auth library.

Distributed Job Scheduler – API Monitoring Platform

Code: github.com/va18hav/distributed-job-scheduler

TypeScript · Node.js · PostgreSQL · Redis · BullMQ · Docker

- Built a distributed job scheduling engine powering a real-world API monitoring platform — clients register a health-check endpoint and a schedule, and the system pings, logs, and alerts on failures automatically.
- Implemented atomic job claiming via conditional row-locking (Postgres UPDATE... WHERE...RETURNING) to guarantee exactly-once execution across concurrent workers, verified under load at **1,500 jobs/sec** with zero duplicate executions.
- Designed a two-stage job pipeline — a recurring ping job and a dependent send_email job — so a failed health check automatically triggers an alert email containing the failure response.
- Built retry logic with exponential backoff and a dead-letter queue, and added a Redis/BullMQ queue layer in front of PostgreSQL to move from polling to push-based job delivery, reducing pickup latency from 1,000ms worst-case (polling) to **5ms** (push-based).

SKILLS

Frontend

React.js Tailwind CSS Context API

Backend

Node.js Express.js REST APIs
WebSockets JWT Authentication
Prisma ORM

Databases

PostgreSQL Supabase Redis

Testing and monitoring

Jest Postman Sentry

Cloud & AI

AWS S3 AWS ECS
AI & LLM Integrations Vercel
Railway Cloudflare Docker

Tools

Git GitHub Claude Code
Google Stitch OpenAI Codex

EDUCATION

B.Tech – Electronics & Communication Engineering

University Visvesvaraya College of Engineering (UVCE), Bengaluru

CGPA: **7.3** / 10.0