Tejas Kale

GitHub | LinkedIn | Portfolio

Location: Parbhani, Maharashtra, India
Email: tejaskale9905@gmail.com

Software Engineer – Backend & Distributed Systems

Software engineer focused on backend and distributed systems, with mobile and web experience. Built a 7-service event-driven commerce backend, an **OpenID Connect** authorization server, and a real-time chat platform with a **React Native (Expo)** app, all as personal projects. Works in **TypeScript/Node.js**, with **C++** for algorithm practice. Writes ADRs for key architectural decisions.

Technical Skills

Languages	:	TypeScript, JavaScript, C++ (DSA practice)
Backend	:	Node.js, Express 5, REST APIs, WebSockets, JWT, OAuth 2.0, OpenID Connect, Zod
Databases	:	PostgreSQL, Drizzle ORM, MongoDB, Redis, Valkey
Infrastructure	:	RabbitMQ, Docker Compose, GitHub Actions, OpenTelemetry, Jaeger
Frontend	:	React, Next.js, React Native (Expo), Tailwind CSS
Testing	:	Vitest, Supertest

Education

Shri Shivaji Science College Parbhani — Maharashtra, India
Bachelor of Computer Applications (BCA) — *Jun 2024 – Expected Jun 2028*

Projects

Flux — *TypeScript, Node.js, PostgreSQL, Valkey, RabbitMQ* — Source Code

- Built a quick-commerce backend as **7 independent services** with a database per service, communicating through RabbitMQ events (payments simulated; runs locally)
- Implemented an event-driven **order saga with compensation**: failed payments release reserved stock, and failed events go to a dead-letter queue
- Built zero-oversell stock reservation with an **atomic conditional UPDATE**: 100 concurrent requests for 1 unit gave exactly 1 success, across 3 replicas
- Added **OpenTelemetry** tracing (one order = one 83-span Jaeger trace), Valkey rate limiting, 96 tests and per-service GitHub Actions CI; wrote 5 ADRs

Grantly — *Express, TypeScript, Drizzle ORM, PostgreSQL, Redis* — Source Code | Live Demo

- Built an **OpenID Connect** authorization server on oidc-provider with a custom Drizzle/PostgreSQL adapter: PKCE, rotating refresh tokens and dynamic client registration (RFC 7591)
- Designed a `/clients` API where users create OAuth apps; secrets are shown once and stored encrypted with AES-256-GCM
- Added Google and GitHub login with account linking, Redis-backed rate limiting, 13 Vitest/Supertest tests and 5 ADRs
- Deployed on Render, Neon and Upstash with a Vercel-hosted dashboard; verified the full login flow end to end

Unsaid (in active development) — *Node.js, WebSockets, Valkey, PostgreSQL, React Native (Expo)* — Source Code

- Built phone-OTP authentication with hashed phone numbers and OTPs, JWT access tokens and rotating refresh tokens
- Implemented real-time **WebSocket** matching (Valkey mood queues) and persistent chat rooms with typing indicators and a heartbeat
- Integrated fail-open AI moderation and AI-generated icebreakers, plus reporting, blocking and bans enforced at login and matching
- Built the Expo client: OTP login, mood picker, real-time chat, report/block actions and profile screen

CollabHub Team Workspace — *React, TypeScript, Vite, Express, MongoDB* — Source Code

- Built a full-stack workspace app: React 19/Vite frontend, Express 5/MongoDB API, JWT auth and Zod validation
- Implemented workspace-based access control, projects, tasks, comments, notifications and activity tracking