

Yash Suthar

Mumbai, India | hello@yashsuthar.dev | linkedin.com/in/yxshsuthar | github.com/trynayash | yashsuthar.dev

EDUCATION

Late Bhausaheb Hiray S.S. Trust's Institute of Computer Applications

Mumbai, MH

Master of Computer Applications (MCA) Sep 2025 – May 2027 CGPA: 8.83/10

Coursework: Data Structures and Algorithms, Operating Systems, Computer Networks, OOP, Distributed Systems, System Design

Parul University

Vadodara, GJ

Bachelor of Computer Applications (BCA) Sep 2022 – May 2025 CGPA: 7.93/10

TECHNICAL SKILLS

Languages: TypeScript, JavaScript, Go, Python, SQL, Java, C, HTML, CSS

Frontend: React, Vite, Tailwind CSS, TanStack Query, Zustand, Radix UI, React Router, Chart.js, Recharts

Backend and Data: Node.js, Express, Go (Gin), REST, GraphQL, WebSockets, BullMQ, PostgreSQL, Redis, MongoDB

DevOps and Testing: Docker, AWS (ECS, RDS, ElastiCache), Terraform, GitHub Actions CI/CD, Vercel, Linux, Vitest, Playwright

EXPERIENCE

[LimitYourAPI](#) — Software Engineer (Founder) limityourapi.tech

Jan 2026 – Present

Go (Gin), React, Redis, Lua, PostgreSQL, Docker, Terraform, AWS, GitHub Actions, Cloudflare Workers, Stripe

- Built and launched an API rate-limiting SaaS end to end: a Go API where every limit is one **atomic Redis Lua** call, a React dashboard, API keys, organizations, TOTP 2FA and **Stripe** billing.
- Measured limit checks at **6–14 μ s of Redis CPU** each (70,000+ checks/s per Redis core); published the Node.js/TypeScript SDK on npm and wrote Go and Python SDKs and a Cloudflare Workers edge limiter.
- Wrote the **CI/CD** in GitHub Actions (Go tests, frontend build, Terraform checks, Docker deploys to **AWS ECS Fargate**) and the AWS infrastructure (RDS, ElastiCache, autoscaling) in Terraform.

[StartupDeckAI](#) — Software Engineer (Founder) startupdeck.in

Jan 2025 – Present

React, TypeScript, TanStack Query, Tailwind CSS, Node.js (Express), PostgreSQL (Drizzle), Redis, BullMQ, GraphQL

- Built a live AI SaaS (214+ users) end to end: **React/TypeScript** UI, Express/GraphQL API, PostgreSQL, Razorpay billing.
- Built its engine: four parallel LLM stages and a synthesis stage feed a **deterministic 8-dimension scoring** formula with a GO / PIVOT / KILL verdict; each figure has a **source label set in code**.
- Moved slow generation into **BullMQ** workers on Redis; lint, type checks, **3,400+ tests** (Vitest, Playwright E2E) and npm audit run in GitHub Actions before each deploy to Vercel and Render.

PROJECTS

PromptP — Real-Time Event-Driven Platform github.com/trynayash/promptp

Jan 2024 – Dec 2024

Node.js, Redis Streams, WebSockets, React

- Made an append-only **Redis Streams** log the single source of truth; every client view is derived from it.
- Added per-socket backpressure and **cursor-resume** reconnects that continue from the client's last event.

CodeFusionAI — Collaborative Coding Environment github.com/trynayash/codefusionai

Jan 2024 – May 2025

Go, Docker, WebSockets, React

- Implemented **Operational Transformation** so several users can edit the same file at once without conflicting changes.
- Ran each session's code in a **Docker** sandbox with cgroup CPU and memory limits.

PUBLICATIONS AND WRITING

Overshoot, Retries and Clocks: An Empirical Study of Redis-Based Rate Limiting Under Concurrency

Oct 2026

Y. Suthar (sole author) · Preprint, Zenodo, not peer-reviewed · doi.org/10.5281/zenodo.23467402 · code and data public

- Tested 8 Redis rate limiters in **3,853 trials**: check-then-act over-admitted 29% at 64 workers, matching a model I derived, $c\text{-ceil}(L/c) - L$; WATCH/MULTI refused up to 42% of valid requests; caller clocks cut a token bucket to 16% of its rate.

Atomic Rate Limiting with Redis and Lua, Measured — Article yashsuthar.dev/writing

Oct 2026

- 7 experiments on Redis 7; e.g. timestamp-keyed sliding logs admitted **12 $\times$** the limit at 10,000 requests/s.