

# Anurag Sharma

Full Stack Developer · AI Integration Engineer · Systems-Minded Builder

[anuragf863@gmail.com](mailto:anuragf863@gmail.com) | +91 83054 42550 | Bhopal, MP 464551, India | [github.com/Anuragsharma15-dell](https://github.com/Anuragsharma15-dell) | [linkedin.com/in/anurag-sharma-22b518330](https://linkedin.com/in/anurag-sharma-22b518330) | [portfolio-seven-sigma-ahvwpqktj.vercel.app](https://portfolio-seven-sigma-ahvwpqktj.vercel.app)

## PROFESSIONAL SUMMARY

Performance-obsessed Full Stack Developer who builds production-grade products end-to-end — from zero-dependency infinite-canvas rendering engines to AI-powered streaming chat platforms. Contributed to open-source (Inkestra frontend, ~2 months). Comfortable owning every layer: React / Next.js 14 frontends, Node / Express backends, PostgreSQL + Supabase data layers, Redis pub/sub, real-time Socket.io pipelines, and OpenAI LLM integrations. Builds from scratch, measures latency, ships fast, and documents every trade-off.

## TECHNICAL SKILLS

<b>Frontend</b>	React, Next.js 14 (App Router), HTML5 Canvas 2D, CSS3, Tailwind CSS, ShadCN UI
<b>Backend</b>	Node.js, Express.js, REST APIs, Server Actions, NextAuth, Rate Limiting, WebSockets
<b>Databases / Cache</b>	PostgreSQL, MongoDB, Supabase, Redis (pub/sub & caching), Prisma ORM, Drizzle ORM
<b>AI / Real-Time</b>	OpenAI API (GPT-4o / GPT-4o-mini), Vercel AI SDK, LLM Streaming, Socket.io
<b>Canvas / Rendering</b>	Infinite Pan & Zoom (affine transforms), Hardware-Accelerated Canvas, HiDPI Export
<b>DevOps / Tooling</b>	Vercel, Git, GitHub, Environment Config, CI/CD, Lighthouse Performance Auditing
<b>Languages</b>	JavaScript (ES2023+), TypeScript, HTML, CSS — Fluent: English, Hindi

## PROJECTS

### Synapse — Real-Time AI Pair-Programming Environment ★

[synapse-dev.vercel.app](https://synapse-dev.vercel.app) (concept)

#### FEATURED

Next.js 14 | Socket.io | Redis Pub/Sub | OpenAI Streaming | Monaco Editor | PostgreSQL / Prisma | Vercel

- Built a browser-based collaborative code editor where multiple developers share a live session — each keystroke synced in <50ms using Socket.io rooms backed by Redis pub/sub, eliminating the fan-out bottleneck that plagues naive WebSocket broadcast architectures.
- Integrated GPT-4o streaming directly into the editor sidebar: AI suggestions stream token-by-token while the user types, achieving a time-to-first-token of ~280ms by using the Vercel AI SDK's ReadableStream pipeline instead of waiting for full completion.
- Implemented Operational Transform (OT) conflict resolution to handle concurrent edits — the same core algorithm used by Google Docs — ensuring no two users overwrite each other's changes even under high concurrency.
- Used Redis as both a session state store (TTL-based room expiry) and a message broker (pub/sub channels per room), decoupling the Socket.io server from the Next.js API layer and enabling horizontal scaling with zero code changes.
- Designed a code execution sandbox using isolated Node.js child\_process workers with a 5s hard timeout and memory cap — users can run JS/TS snippets in-browser without exposing the host.
- Tracked room analytics (active users, edits/min, AI call count) in PostgreSQL via Prisma, surfaced in a real-time dashboard that updates via Server-Sent Events without polling.

## Pencil — Infinite Canvas for Technical Diagrams

[pencil-sketchpad.vercel.app](https://pencil-sketchpad.vercel.app)

React | HTML5 Canvas 2D | TypeScript | Custom Rendering Engine

- Reverse-engineered the architecture of Excalidraw-class tools and built an infinite canvas from scratch — zero third-party canvas libraries, every primitive hand-rolled.
- Implemented smooth pan & zoom using affine transformation matrices, batching dirty-region redraws to maintain interaction latency under 16ms (60fps) on large, complex diagrams.
- Designed an element-based rendering pipeline where every shape is an independent serialisable object, enabling undo/redo, multi-select, and per-element style overrides without full canvas invalidation.
- Built a precision dual-grid system (20px minor / 100px major) with snap-to-grid logic — a missing feature in most open-source diagram tools that Pencil solves natively.
- Hardware-accelerated rendering via requestAnimationFrame scheduling; HiDPI PNG export uses device-pixel-ratio scaling for crisp output at any zoom level.

## Breeze — Production AI Chat Platform

[breeze-ochre.vercel.app](https://breeze-ochre.vercel.app)

Next.js 14 | OpenAI GPT-4o-mini | Supabase / PostgreSQL | NextAuth (Google OAuth) | Prisma | Vercel

- Architected a full-stack AI chat app end-to-end: streaming responses, persistent history, favourites, auto-generated session titles, and per-user rate limiting — all deployed on Vercel.
- Reduced perceived latency from ~2s (full completion) to <300ms time-to-first-word by streaming GPT tokens via Vercel AI SDK's useChat hook with a ReadableStream response pipeline.
- Server-side rate limiting (RATE\_LIMIT env var) enforced via Supabase row-level checks, blocking abuse without adding a Redis dependency and keeping cold-start times minimal.
- Used Next.js Server Actions to auto-generate GPT-powered session titles on first message, replacing the generic 'New Chat' UX seen in most open-source clones.
- Versioned REST API (/api/v1/chat, /rate-limit, /favourite) cleanly decoupled from the UI layer; Google OAuth + NextAuth session management with zero token leakage.

## OPEN-SOURCE CONTRIBUTIONS

---

### Inkestra — Open-Source Platform (Frontend Contributor)

~2 months active contribution

React | Tailwind CSS | Open-Source Collaboration | GitHub PR Workflow

- Contributed frontend features and UI improvements to Inkestra, an open-source platform, over approximately two months — working within a real collaborative codebase with PR reviews, branch management, and community coding standards.
- Gained hands-on experience with production open-source workflow: issue triage, feature branches, pull request feedback cycles, and merging contributions into a shared main branch.

## EDUCATION

---

### B.Tech — Computer Science & Engineering

2022 – 2026 (Expected)

Sagar Institute of Research and Technology (SIRT) · Bhopal, MP, India

## LANGUAGES

---

**English** — Professional fluency · **Hindi** — Native speaker

## WHAT MAKES ME DIFFERENT

---

- Builds from scratch first — understands every abstraction before reaching for a library.
- Performance-driven: measures latency with DevTools and Lighthouse before calling any optimisation done.
- AI-native developer: integrates LLMs as first-class features with streaming, prompt engineering, and cost/latency trade-offs baked in.
- Full ownership: designs, builds, deploys, and monitors — no hand-offs mid-feature.