

Nikhil Sharma

+91 81465 18102 | nsharma5_be24@thapar.edu | github.com/niksharma

EDUCATION

Thapar Institute of Engineering and Technology

B.E. Computer Science and Engineering | **CGPA: 9.95 / 10**

2024 – 2028

Patiala, Punjab, India

PROJECTS

Relay | *React, TypeScript, Node.js, Express 5, PostgreSQL, Prisma, native ws, JWT* | [GitHub](#)

- Built a full-duplex chat app on **raw WebSockets** (no Socket.IO) — manually handling message framing, connection lifecycle, and protocol design using the browser's native WebSocket API
- **Dual-protocol architecture**: REST for auth and message history while WebSockets handle low-latency real-time delivery
- Engineered a **SocketManager** class supporting multi-device connections, presence tracking, and broadcast patterns — replacing a naive in-memory Map with a proper connection lifecycle model
- Solved WebSocket auth by **manually parsing the cookie header** from the raw HTTP upgrade request; implemented **Zod validation** across all REST endpoints and WebSocket payloads
- Shipped **message delivery status** (Sent → Delivered → Read) via a bidirectional acknowledgment state machine
- Real-time **typing indicators** with debounced WebSocket events
- Tracked down a **stale closure bug** in the *useWebSocket* hook causing messages to appear in the wrong chat window — fixed using *useRef* to always hold the latest callback

Wallet System | *Next.js, Express.js, PostgreSQL, Prisma, NextAuth, Docker, GitHub Actions, TurboRepo* | [GitHub](#)

- Prevented race conditions in concurrent transfers using **optimistic concurrency control** with database-level conflict detection — ensuring no double-spend under simultaneous requests
- Designed and deployed a full **CI/CD pipeline** with GitHub Actions: builds a Docker image on push, publishes to Docker Hub, and deploys to an **AWS EC2** instance automatically
- Built an Express-based webhook service to process async banking events and keep transaction state consistent between services
- Managed as a **TurboRepo monorepo** — shared TypeScript types and configs across Next.js frontend and Express backend with a single lockfile

Medium-like Blogging Platform | *React, Node.js, Express, PostgreSQL, Prisma, JWT, Docker Compose* | [GitHub](#)

- Published a **shared npm package** containing Zod schemas and TypeScript types — consumed by both frontend and backend for consistent runtime validation without duplication
- Containerised the full stack with **Docker Compose** for one-command local setup across frontend, backend, and database services
- Implemented JWT + bcrypt authentication with secure backend APIs and a responsive Vite + TailwindCSS frontend

TECHNICAL SKILLS

Languages: TypeScript, JavaScript, Python, SQL, Go (learning)

Frontend: React.js, Next.js, TailwindCSS, Vite, HTML, CSS

Backend: Node.js, Express.js, REST APIs, WebSockets, JWT, NextAuth, bcrypt

Databases: PostgreSQL, MongoDB, Prisma ORM — indexing, transactions, concurrency control

DevOps: Docker, Docker Compose, GitHub Actions (CI/CD), AWS EC2, Nginx, Redis

Tools: Git, TurboRepo, Zod, pnpm, k6 (load testing)

INTERESTS

Formula 1 · Reading · Exploring Emerging Technologies