

# Yashraj Yadav

Full-Stack Developer

+91-9665572020 yashrajyadav4660@gmail.com [LinkedIn](#) [GitHub](#) [Portfolio](#)

## Summary

---

Self-taught full-stack developer building production-ready systems with TypeScript, Next.js, and PostgreSQL. Focused on AI-integrated products, transactional systems, and real-time applications.

## Skills

---

**Frontend & Backend:** React.js, Next.js, Node.js, Express.js, Socket.io, REST APIs, TypeScript, JavaScript, Tailwind CSS, Framer Motion, GSAP

**AI:** Gemini API, Vercel AI SDK, LangGraph, LangChain, RAG, OpenAI, Anthropic

**Databases & Tools:** PostgreSQL, MongoDB, SQLite, Drizzle ORM, Prisma, Git, GitHub, Vercel, Render, Bun

## Projects

---

- **Zeno Bank (Banking Ledger)** GitHub | Live  
*Next.js, TypeScript, PostgreSQL, Drizzle ORM*
  - Architected a **double-entry ledger system** enforcing strict debit/credit invariants for financial correctness.
  - Designed **atomic SQL transaction workflows** with rollback safety, ensuring consistency under concurrent requests.
  - Implemented **idempotency keys** to eliminate duplicate transactions and prevent double-spend scenarios.
  - Derived account balances from ledger entries (no stored state), ensuring **auditability and tamper resistance**.
  - Validated correctness across **100+ concurrent transaction simulations**.
- **Private Rooms — Ephemeral Chat** GitHub | Live  
*Next.js, TypeScript, Socket.io, WebSockets*
  - Engineered a **WebSocket-based real-time messaging system** achieving sub-100ms latency under concurrent usage.
  - Designed an **ephemeral architecture** with automatic room lifecycle management (creation → expiry → deletion).
  - Enabled anonymous communication via **short-lived room codes**, minimizing user friction.
  - Implemented **rate limiting and connection controls** to prevent spam and stabilize server load.
- **SQL AI Agent** GitHub | Live  
*Next.js, TypeScript, Gemini API, SQLite*
  - Built a **natural language → SQL execution engine** enabling real-time querying over structured data.
  - Engineered **schema-aware prompt pipelines** to generate contextually correct and executable SQL.
  - Implemented **query safety guardrails** (read-only enforcement, validation layers) to prevent destructive operations.
  - Reduced manual query overhead by automating common retrieval workflows via NL interfaces.
- **Foliokit** GitHub | Live  
*Next.js, TypeScript, Tailwind CSS, Framer Motion*
  - Built a **portfolio template platform** with reusable, structured UI systems.
  - Designed **pre-configured starter kits** to accelerate developer onboarding and setup.
  - Developed **interactive UI components** using motion-driven layouts (bento grids, transitions).
  - Reduced portfolio setup complexity by standardizing layout, components, and design patterns.