Prateek Panwar

Gurugram, India • work@prateekpanwar.com
prateekpanwar.com • linkedin.com/in/prateek72 • github.com/kzekiue

Full-Stack Software Engineer, Web Applications, Data & Automation

Summary

Full-stack engineer with 3+ years building TypeScript/JavaScript frontends, Java/Python/Node backends, and PostgreSQL/SQLite data systems. Redesigned a fintech document pipeline for a 60x runtime cut (40 min → 40 s) and built ingestion services across SFTP, CSV, and Excel sources. Strong fit for internal tools, automation, debugging, data modeling, and reliable web systems.

Technical Skills

Languages: TypeScript, JavaScript, Python, SQL, Java
Frontend: React, Next.js, Angular, Stencil.js, Tailwind CSS, D3.js
Backend: FastAPI, Spring Boot, Node.js, Bun, REST, WebSockets
Data: PostgreSQL, SQLite, Oracle SQL, Redis, SQLAlchemy, Drizzle ORM, CSV/Excel ingestion
Infra/Testing: Docker, GitHub Actions, Nginx/Caddy, systemd, AWS, GCP, Vitest
AI Tooling: MCP, Gemini API, Tool-backed AI Workflows, LLM-Assisted debugging, Background jobs

Experience

Associate Software Engineer, Product *Jul 2023 – Present*
Lentra AI, Noida, India

- Redesigned the co-lending document pipeline with multithreaded processing, cutting runtime from ∼40 min to ∼40 s (60x) and compressing PDF/image payloads from ∼90 MB to a few MB; built SFTP auto-ingestion, strict file validation, and resilient CSV upload flows. *(Angular, Spring Boot, PostgreSQL, Redis, SFTP)*
- Built a batch ingest service processing 1,000+ records per batch, aggregating Excel and CSV sources, plus a backend expression parser evaluating dynamic financial formulae at runtime. *(Spring Boot, Angular, PostgreSQL)*
- Architected a Stencil.js Web Component library for the Business Rule Engine from scratch: reusable wrappers, TypeScript interfaces, and coding standards adopted team-wide; mentored two interns from Figma handoff to delivery.
- Delivered LMS dashboards, a real-time chat interface, D3.js loan-journey visualizations, and Razorpay integration for NBFC fund collection; drove an Angular version upgrade for performance and maintainability.

Full-Stack Developer, Intern *Sep 2022 – Jun 2023*
Lentra AI, Noida, India

- Built the LMS frontend from the ground up; the dynamic-list and List-of-Values components were adopted across multiple products and remain in production. Implemented system-wide theming, earning a **Spot Award**.
- Drove an Angular migration off a legacy version; contributed across the full SDLC from requirements gathering through production deployment. *(Angular, Java, PostgreSQL)*

Selected Projects

Kosh: Self-hosted personal-finance platform *TypeScript, Next.js, PostgreSQL, Drizzle*
github.com/kzekiue/kosh

- Full-stack ledger platform (accounts, CSV imports, budgets, bills, goals, reports, recurring transactions) with a Turborepo split across domain logic, Drizzle/PostgreSQL persistence, and Next.js application delivery.
- Designed correctness safeguards in SQL and code: integer minor-unit money, transaction-sign constraints, ownership checks, partial unique indexes for duplicate-import detection with soft delete.
- Implemented AES-256-GCM column encryption, scoped/hashed read-only MCP tokens, pg-boss background jobs, and backup/restore/migration-verification tooling; **137 passing tests**.

Scoutexa: B2B lead-discovery & CRM platform *(in progress)* *Next.js, Python, FastAPI, PostgreSQL*

- Built a mixed TypeScript/Python monorepo: Next.js feature-module UI and a FastAPI service layer over async SQLAlchemy/asyncpg on PostgreSQL (Supabase).
- Built a bounded enrichment pipeline aggregating Google Places, Firecrawl, and self-hosted SearxNG, crawling likely contact/team pages, normalizing emails and E.164 phones, and preserving per-channel source, confidence, and verification metadata.
- Added an evidence-constrained Gemini resolver that rejects model outputs absent from collected source data, plus a retryable PostgreSQL job queue with worker concurrency and request latency/query-count metrics.

TinyReplay: Self-hosted browser session-replay tool *TypeScript, Next.js, rrweb, SQLite*
github.com/kzekiue/tinyreplay

- Built end-to-end: browser SDK (rrweb) → validated ingest API → SQLite storage → custom replay dashboard; packaged with Docker and published through GHCR.
- Designed idempotent ingestion using stable batch IDs, recorder lifecycle ordering, bounded buffers, and unload-time sendBeacon; capture-time input masking keeps sensitive data in the browser.
- Added SQLite migrations (PRAGMA user_version, WAL), retention sweeps, rate limiting, CORS/token controls, and CI with a container smoke test.

Koguko: Live anonymous real-time chat *Next.js, Bun WebSockets, SQLite*
koguko.com

- Built and deployed a text-first random-chat app on a native Bun HTTP/WebSocket server with FIFO matchmaking and no ordinary chat-history retention.
- Designed an ephemeral-by-default data boundary: normal chats stay in memory; reports and severe safety flags persist bounded context to an encrypted (AES-GCM) SQLite moderation plane with hashed-IP bans and retention sweeps.
- Implemented connection resilience (typed discriminated-union messages, heartbeat/PONG, reconnect, graceful drain) and AWS/Caddy/systemd deployment with WAL-checkpointed SQLite backups; **231 backend tests**.

Open Source

- **opencode** (open-source AI coding agent): Root-caused a bug, filed a reproducible issue, and shipped the fixing PR.
- **Visual Studio Code**: Investigated a desktop window-reload bug and proposed a code-level fix.