

DNYANESHWAR KAWADE

[LinkedIn](#) | [Portfolio](#) | [GitHub](#) | [+91 9067183889](#) | dnyaneshwarpk.07@gmail.com

Backend engineer building fault-tolerant distributed systems and scalable multi-tenant architectures. Experienced in designing resilient asynchronous pipelines, zero-downtime data migrations, and high-performance APIs with full end-to-end ownership. Stack: Node.js, TypeScript, Redis, Docker, AWS.

EXPERIENCE

WellnessZ, *Full Stack, Remote*

Aug 2024 – Present

- Shipped a multi-tenant SaaS platform from scratch — 150+ REST endpoints in 25 days — with tenant-isolated architecture and zero cross-tenant data leaks in production.
- Reduced peak database load by 80% by introducing Redis/BullMQ async job processing, eliminating primary-thread blocking during traffic spikes.
- Executed a zero-downtime schema migration merging disparate web and mobile user collections — zero data loss, no user-facing downtime.
- Integrated 3+ third-party services (Razorpay, YouTube, Zoom) via fault-tolerant async pipelines with zero transaction drift.
- Owned the multi-tenant isolation model end-to-end, then handed off implementation to a developer I mentored.

PROJECTS

Vibhava Estate Systems — Multi-Tenant Real Estate SaaS

Node.js, TypeScript, Next.js, Express.js, Redis, BullMQ, Docker, AWS

- Designed multi-tenant backend with isolated tenant data, dedicated API service, and decoupled worker service for async processing.
- Built CI/CD workflows (GitHub Actions) with containerized Docker Compose deployments to AWS EC2.
- Implemented async task processing with Redis to offload long-running jobs from main API; integrated Redis for job queuing and caching.

Distributed Event Bus & Notification Engine

TypeScript, Node.js, Express.js, Redis, BullMQ, Docker, AWS

- Achieved sub-2ms API ingestion latencies by decoupling event architecture with Redis primitives, eliminating primary-thread DB blocking and cutting endpoint overhead by 95%.
- Engineered real-time BullMQ fan-out pipeline partitioning push, SMS, and email dispatches into isolated parallel worker pools, cutting processing overhead by 60% with 99.9% delivery reliability.
- Containerized worker architecture using Docker for horizontally scalable channel queues, preventing async traffic spikes from impacting core platform performance.

SKILLS

Primary

Node.js · TypeScript · JavaScript · Go · Express.js · Redis · BullMQ · Docker · AWS

Secondary

React.js · Next.js · Redux Toolkit · REST APIs · Microservices · Multi-Tenant Architectures · Asynchronous Workflows · GitHub Actions CI/CD · Git · GitHub

EDUCATION

Bachelors in Computer Science — VIIT, Pune

July 2021 – May 2025

Relevant Coursework: Databases, Operating Systems, Computer Networks, DSA

ACHIEVEMENTS

- 1st place, Valiant 2023 — Vishnu Institute of Technology, Kovvada. Built a plant disease detection system in a 36-hour hackathon.