

Bhargav Deshpande

Pune, Maharashtra, India | bhardesh2005@gmail.com | +91 94030 18661

[Portfolio](#) | [LinkedIn](#) | [GitHub](#)

Education

MIT World Peace University — Pune, Maharashtra

Aug 2023 – May 2027

B.Sc. in Data Science & Big Data Analytics / GPA: 8.2/10

Relevant Coursework: Data Structures & Algorithms, Database Management Systems, Distributed Computing, Object-Oriented Programming, Operating Systems, Probability & Statistics, Machine Learning

Experience

Servzo (VendorGrid B2B Procurement Platform) | Co-Founder & Full-Stack Engineer

Jan 2025 – Present

Pune, Maharashtra

- Improved backend throughput by **28%** (183 → 235 req/s) by profiling and eliminating worker-process context-switching overhead — tuning a Gunicorn + asyncpg stack to a single-worker async model verified with **wrk** load testing at 10–100 concurrent connections, sustaining **sub-100 ms median latency** on a 1 vCPU / 4 GB VPS.
- Engineered a **high-concurrency monorepo backend** with FastAPI, asyncpg, and SQLAlchemy 2.0 (async), applying object-oriented design patterns and non-blocking event loops to support concurrent bidding sessions with no thread-safety issues.
- Replaced WebSocket broadcast system with a **server-sent events (SSE)** fan-out architecture backed by an async Redis queue, eliminating full-duplex overhead and reducing connection-state complexity by ~40%.

AWS Cloud Club — MIT-WPU | Software Developer

Sep 2024 – Jan 2025

Pune, Maharashtra

- Designed and implemented a **genetic algorithm** timetable scheduler in Python, optimising a multi-constraint fitness function from 47% → **98% fitness score** across 2,000 evolutionary iterations, reducing manual scheduling effort by an estimated 6 h/week.
- Authored a **CLI automation toolkit** leveraging Python **asyncio** for concurrent task orchestration, cutting peer setup time by **60%** (~45 min → ~18 min) across 30+ club members.

Projects

Watchtower — Real-Time Observability Dashboard | [GitHub](#)

Summer 2026

- Built a **self-hosted log ingestion pipeline** (Elysia on Bun, Redis, ClickHouse) with a Redis → SSE → React hot path; removed a blocking **await** from the ingest handler and raised throughput from <500 req/s to a **sustained 7,251 req/s** (peak 8,595 req/s) at **0% error rate**, p50=42 ms / p99=76 ms, verified at 512 concurrent connections.
- Implemented **end-to-end type safety** across the monorepo using Elysia Eden Treaty, eliminating a class of runtime API contract errors at compile time.
- Authored a **postmortem blog post** on diagnosing the 4× throughput regression — a debugging write-up covering the fix and the trade-offs made.

Vriddhi — NSE End-of-Day Data Pipeline | [GitHub](#)

Summer 2026

- Built a **self-hosted NSE data pipeline** and archive covering twenty years of market data; downloads, parses, and bulk-loads CM and F&O bhavcopies into PostgreSQL using asyncpg binary COPY protocol.
- Architected a strict one-way dependency flow (**core** → **db** → **ingestion** → **api**) with a **FastAPI REST API** and Redis-backed task queue (**arq**) for orchestration.
- Developed a **React screener dashboard** with shadcn/ui and TanStack Query, consuming typed API hooks generated by Orval from the backend OpenAPI schema.

Technical Skills

Languages	Python (Advanced), TypeScript, C, SQL
Backend & Systems	FastAPI, asyncio, asyncpg, SQLAlchemy 2.0, BullMQ, Elysia (Bun), Redis Pub/Sub, REST, SSE, WebSockets
Frontend	React, TanStack (Query / Router / Start), Zustand, Pydantic v2, OpenAPI / orval
Databases	PostgreSQL, ClickHouse, Redis
Infrastructure	Docker Compose, Nginx, Let's Encrypt, DigitalOcean, AWS (EC2, S3), Git, rsync/SSH
Tooling	wrk / load testing, Alembic (migrations), uv (Python packaging), Bun