Jay Bhende

Full Stack Engineer

Mobile: +91-9325712924
Email: bhendejay5.bms@gmail.com

Github: JayBhende05
LinkedIn: Jay Bhende

Summary

Full-stack engineer building event-driven web systems with TypeScript, Node.js, Next.js, Python and PostgreSQL. Projects include a crypto exchange with a custom matching engine, workflow automation with a transactional outbox and Kafka workers, a support platform with race-safe approval workflows, and a digital wallet with locked-balance handling. Three client websites in production since 2025.

EDUCATION

- **Master of Computer Application (MCA)** *(Savitribai Phule Pune University)* — Maharashtra, India
  ZIBACAR College, Pune CGPA: 8.5 — *June 2024 - June 2026*

- **Bachelor of Computer Application (BCA)** *(Savitribai Phule Pune University)* — Maharashtra, India
  S.N. Arts, D.J. Malpani & B.N. Sarada College, Sangamner CGPA: 8.5 — *June 2020 - June 2024*

SKILLS SUMMARY

- **Languages**: JavaScript, TypeScript, SQL, HTML/CSS, Python
- **Frontend**: React, Next.js, Tailwind CSS, TanStack Query, React Flow, GSAP, Framer Motion
- **Backend**: Node.js, Express.js, FastAPI, REST APIs, WebSockets, Prisma, SQLAlchemy
- **Testing**: Jest, Vitest
- **Database**: MongoDB, PostgreSQL, MySQL, Redis
- **Deployment :** Vercel, EC2, Nginx, Docker Compose
- **DevOps/Tools**: Git, GitHub, Docker, AWS (Cloud Practitioner), Linux, VS Code
- **AI/LLM**: LangGraph, tool-calling agents, RAG, hybrid retrieval (vector + BM25), RRF, cross-encoder reranking, structured outputs (Pydantic), human-in-the-loop approval, Langfuse and OpenTelemetry tracing, Anthropic, Gemini and Groq APIs, ChromaDB

Certification

- **MERN Stack Application Development** – TechSaksham (Microsoft & SAP CSR Initiative, implemented by Edunet Foundation)
  Completed hands-on training in full-stack web development using MongoDB, Express.js, React.js, and Node.js.

- **AWS re/Start Cloud Practitioner Program** — Generation India
  Covered AWS Cloud Foundations, Linux, Networking, Security, Python basics, Databases, Web Servers, and Cloud Infrastructure concepts.

Experience

Freelance Full-Stack Developer
Self-Employed | Remote
2025 – Present

- Built and deployed 3 production websites with Next.js, React, TypeScript and Tailwind CSS:
  - Aquila Marketers (digital marketing agency)
  - Snappy Motions (motion design studio)
  - Kintsugi Village –((early-learning centre, including an admin access system).)
- Designed responsive and animation-rich user interfaces using GSAP and Framer Motion.
- Managed deployment workflows, performance optimization, CDN integration, and production maintenance.
- Collaborated directly with clients to understand requirements, implement features, and deliver scalable web solutions.

PROJECTS

- **Triage Agent - Human-in-the-Loop AI Support Agent** (Python, FastAPI, LangGraph, Pydantic, PostgreSQL, SQLAlchemy, React, TypeScript, Docker)
  Github :
  Built a **full-stack support-ticket platform** where an AI agent proposes actions (reply, refund, close, or escalate), **deterministic business rules control execution**, and reviewers approve risky actions from a web dashboard.
  - Built a **FastAPI backend** with REST endpoints for tickets, approvals and stats, background job execution, a uniform error format, and **string-based money handling** to prevent floating-point precision errors.
  - Ensured **concurrency safety and idempotent refunds** with conditional UPDATE for ticket claiming, SELECT ... FOR UPDATE on approval decisions, and unique keys with row locks, so double clicks return **HTTP 409** instead of creating duplicate refunds.
  - Built a **durable approval workflow** by checkpointing state in PostgreSQL. Paused tickets resume after server restarts, and decisions recorded just before a crash are preserved without requiring reviewers to approve them again.
  - Developed a **React dashboard with TanStack Query** showing the approval queue, ticket evidence, editable proposals and step-by-step execution traces. **TypeScript types generated from the OpenAPI schema** ensure breaking backend changes fail the frontend build.
  - Integrated an **LLM agent restricted to read-only tools**, with deterministic risk rules in Python and an **audit trail tracking latency and cost**. Tested with pytest against a real PostgreSQL database and with Vitest.

- **Trado - Real-Time Cryptocurrency Exchange Platform** (TypeScript, Node.js, Next.js, Redis, WebSockets, TimescaleDB, Binance APIs)
  Github :
  Built a real-time cryptocurrency exchange platform where users can place buy/sell orders, with a custom matching engine, asynchronous trade processing, and live order book and market updates.
  - **Engineered a cryptocurrency exchange from scratch** with a modular architecture comprising an API Server, in-memory Matching Engine, WebSocket Server, and asynchronous Database Processor.
  - Implemented **event-driven communication with Redis**, decoupling order processing, trade persistence, and real-time market-data distribution.
  - Developed a **real-time WebSocket infrastructure** for streaming incremental order-book depth, trades, and market events to connected clients with subscription management.
  - Implemented **asynchronous trade persistence using TimescaleDB**, keeping database operations off the critical order-execution path; integrated Binance REST/WebSocket APIs for initial market-data simulation.

- **TriggerLab - Event-Driven Workflow Automation Platform** (TypeScript, Node.js, PostgreSQL, Prisma, Kafka, Webhooks, Solana)
  Github :
  Built an event-driven workflow automation platform where users can create webhook-triggered workflows and execute actions asynchronously through Kafka-based workers, including email and Solana transfers.
  - **Designed and built an event-driven workflow automation platform** with independent API, webhook, database-processing, and worker services.
  - Implemented an **Outbox-based asynchronous execution pipeline**: Webhook → Zap Run → Outbox → DB Processor → Kafka → Worker, decoupling event ingestion from action execution.
  - Built **Kafka workers with retry handling and offset management**, enabling failed actions to be reprocessed without losing workflow events.
  - Implemented **ordered multi-action workflow execution**, where each compl eted action publishes the next action to Kafka for independent worker processing.
  - Built extensible workflow execution supporting **dynamic inputs from webhook payloads**, with Email, Discord and Solana transfer actions and architecture designed for additional integrations.

- **Monexo - Digital Wallet & Payment Platform** (TurboRepo, Next.js, NextAuth.js, TypeScript, Prisma, PostgreSQL)
  Github :
  Monexo is a full-stack fintech wallet platform that allows users to securely deposit, withdraw, and transfer money through a simulated banking infrastructure using atomic transaction handling.
  - Implemented **user/merchant authentication, Google OAuth, wallet management, deposits, withdrawals, P2P transfers, and transaction history**.
  - Designed **locked-balance handling for withdrawals** and atomic balance updates using Prisma transactions to maintain financial consistency during concurrent operations.
  - Built a dedicated **Bank Handler service** to simulate external banking operations and separated financial responsibilities across services.
  - Developed a **TurboRepo-based multi-app architecture** to organize the wallet frontend, backend services, and shared packages.
  - Focused on preventing **race conditions, inconsistent balances, and partial transactions** across wallet and banking operations.

ACHIEVEMENTS

- **Developed** a full-stack event management system for **TechMantra 2025**, handling **450+ registrations** and **₹20K+ transactions** with **real-time admin tracking** using **React, Node.js, and MongoDB**.
- Published a paper in **IJSREM Journal (Vol. 09)** on **"Impact of IPv6 Adoption on Internet Infrastructure"**
- **Ranked 1st** in Student Research Project presentation on **"Use of AI in Environmental Sustainability"**