

Jay Bhende

Full-Stack Engineer

Email: bhendejay5.bms@gmail.com

Github: [JayBhende05](#)
LinkedIn: [Jay Bhende](#)

Summary

Full-stack engineer building real-time, event-driven and LLM-powered applications with TypeScript, Node.js, Next.js, Python and FastAPI. Projects include a crypto exchange with a custom matching engine, a Kafka-based workflow automation platform, and a RAG system with benchmarked hybrid retrieval and validated citations.

EDUCATION

• Master of Computer Application (MCA) (<i>Savitribai Phule Pune University</i>) ZIBACAR College, Pune CGPA: 8.5	Maharashtra, India June 2024 - June 2026
• Bachelor of Computer Application (BCA) (<i>Savitribai Phule Pune University</i>) S.N. Arts, D.J. Malpani & B.N. Sarada College, Sangamner CGPA: 8.5	Maharashtra, India June 2020 - June 2024

SKILLS SUMMARY

- **Languages:** JavaScript, TypeScript, SQL, HTML/CSS, Python
- **AI/LLM:** RAG, hybrid retrieval (vector + BM25), RRF, cross-encoder reranking, ChromaDB, Gemini API, structured output, retrieval evaluation
- **Frontend:** React.js, Next.js, NextAuth.js, Tailwind CSS, GSAP, Framer Motion, Recoil, Context API, Redux
- **Backend:** Node.js, Express.js, Hono, FastAPI, REST APIs, gRPC, WebRTC, Authentication, Prisma, Zod
- **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

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

- Developed and deployed production-grade business websites using Next.js, React.js, TypeScript, and Tailwind CSS.
- **Built and maintained live client platforms** including:
 - [Aquila Marketers](#) - Premium digital marketing agency platform with immersive animations and story-driven UI.
 - [Snappy Motions](#) - Creative studio website optimized with Bunny.net CDN and media delivery infrastructure.
 - [Kintsugi Village](#) - Preschool and educational platform focused on modern UI/UX and responsive design.
- 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.

- **ResearchPilot - Grounded RAG System for Research Papers (Python, FastAPI, ChromaDB, BM25, BGE Reranker, Gemini API, Next.js, TypeScript)**

[Github](#) :

Built a retrieval-augmented generation system that answers questions over technical research papers and returns the validated evidence behind each answer, developed across four benchmarked versions.

- * Implemented **hybrid vector + BM25 retrieval with RRF fusion** and **BGE cross-encoder reranking**, raising evidence **Top-3** from **70% to 90%** and **Top-5** from **80% to 92.5%** while holding **document retrieval at 100%**.
- * Built a **40-question benchmark across 4 papers** with document-level and page-level evidence labels, replacing subjective checks with Top-1/3/5 retrieval metrics.
- * Ran **8 controlled retrieval experiments** (chunking, query transformation, multi-query, hybrid search, reranking, pool size) and kept only the 2 that measurably improved results.
- * Added grounded generation with **Gemini structured output and citation validation**, so answers can only cite chunks the retriever supplied.
- * Served the pipeline through a **FastAPI** service with models loaded once at startup and **per-stage latency reporting**, and built a **Next.js** chat UI with a sources panel.

- **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.
- * Built a **Matching Engine** supporting order-book management, buy/sell order matching, trade execution, balance updates, and fill generation.
- * 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 completed action publishes the next action to Kafka for independent worker processing.
- * Built extensible workflow execution supporting **dynamic inputs from webhook payloads**, with Email 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 **IJSERM 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”**