Paramjeet Singh

India • paramjeetsinghpawar25@gmail.com • (+91)7734050945
• www.linkedin.com/in/paramjeet-singh-b566b036b

SUMMARY

Full-stack software engineer with **~1 years** of experience building and scaling **Full-stack Applications, Trading tools, AI tools** and **SaaS products** across backend **architecture, infrastructure** and implementation-focused on **distributed systems**.

WORK EXPERIENCE

Full-stack Software Engineer — Nov 2025 – present

100xDevs — **Delhi NCR, India (Remote)**

- Led the end-to-end design and development of a high-performance algorithmic trading platform built on a low-latency C++ core with a modern React-based terminal. Engineered a distributed, microservices-style architecture with WebSocket-based real-time data streaming, a multi-threaded datafeed system, and a high-speed execution engine with integrated risk management. Implemented hybrid execution modes for live trading and historical backtesting, along with an embedded browser-based IDE for strategy development, real-time visualization, and seamless strategy deployment.
- Architected a simulated crypto CFD trading engine (BTC, ETH, SOL) as an event-driven microservice system wired through Redis Streams and questDB snapshots, so trades replay deterministically, positions survive restarts, and the matching path stays lowlatency and horizontally scalable under load.
- Built a real-time portfolio tracker using TypeScript with Redis caching, precomputed PnL metrics, and load-balanced WebSocket pipelines for low-latency updates.

Full-stack Software Engineer Intern — Jun 2025 – Nov 2025

Peples Brands Lab — **Delhi NCR, India (Remote)**

- Drove a ecommerce SaaS from idea to MVP in ~3 months by designing a scalable backend + frontend architecture, implementing feed/cart/order/payment flows, and building a reusable Next.js + Tailwind component library that replaced spreadsheet workflows and cut new dashboard view build time from days to hours.
- Partnered with founders from zero to launch to ship the first production website, implementing a modular component system (hero, feature rows, FAQ, CTA) and wiring in analytics and env-based config so the team could roll out, A/B test, and rollback new pages safely within days instead of weeks.

PROJECTS

Blink – Backtesting Engine

- Built a low-latency, event-driven quantitative research and backtesting platform in C++, enabling users to develop strategies, run simulations on historical datasets, and execute in live markets with real capital
- Designed for scalability using load balancing, memory replication, and multi-server architecture to ensure fault tolerance, high availability, and distributed processing
- Utilized Postgres for high-performance time-series market data, providing ultra-low latency ingestion, high throughput, and efficient analytical queries,
- Leveraged Boost.Asio and Boost.Beast for fine-grained control over the networking layer, enabling efficient asynchronous I/O, high-throughput data streams, and low-latency system performance
- Built a React + TypeScript frontend providing an interactive environment for strategy development, visualization, and real-time monitoring

Jinx – Hight Frequency Trading Platform

- Built an ultra low latency, event driven high frequency trading system in C++, leveraging Boost.Asio and Boost.Beast for efficient asynchronous socket I/O, enabling fine grained data flow control and high throughput stream processing.
- Implemented a replicated in-memory order book for fault tolerance and consistent state management, enabling seamless failover, and used a lock-free event queue to process market events with minimal contention.
- Consumed market events from a lock-free queue and applied trading strategies, with a modular design allowing future integration with hardware acceleration e.g., FPGA, and routing
- Only risk-validated orders using pre-trade checks (position sizing, margin, leverage limits, stop-loss, trading session) and microstructure models (order book imbalance, microprice) to avoid adverse trades in fast markets.

Merchant dashboard

- Built a scalable, real-time merchant analytics platform processing ~10K events per minute, designed with a modular monolith architecture that cleanly separates ingestion, processing, and serving layers—ready to evolve into a fully event-driven system.
- Backend is stateless NestJS, deployed behind an Nginx load balancer for horizontal scaling across multiple instances.
- PostgreSQL serves as the source of truth for transactional data, while Redis provides a low-latency caching layer.
- Kafka decouples high-volume event ingestion from processing, ensuring resilience to traffic spikes and preventing backpressure on the API.
- A dedicated precompute worker consumes events asynchronously, maintains aggregated metrics, and stores them for O(1) dashboard read queries—delivering sub-500ms analytics responses at scale.
- The design prioritises high throughput, fault tolerance, and horizontal scalability, with conscious trade-offs including eventual consistency and added infrastructure complexity.

SKILLS

Computer Science

- Programming Languages: C/C++, Python, Typescript, Javascript
- System Design & Infra: Microservices, Kafka, Distributed Systems, Event-Driven Systems, REST APIs, WebSockets
- Backend & APIs: Boost, Boost.beast, Boost.Asio, REST APIs, WebSockets
- DevOps & Cloud: Docker, Kubernetes, AWS
- Databases & Caching: MySQL, PostgreSQL, QuestDB, Redis, MongoDB
- Frontend: React.js, Next.js, Dear ImGui, Vulkan Api, Tailwind CSS

Trading and Finance

- Research: Jupiter Notebook, Quant Connect, Meta trader 4/5
- Exchanges & Brokers: Coindcx, Zerodha, Grow, Upstocks, Binance, Bybit, Delta Exchange, Exness
- Proprietary Trading: FundedNext, Blueberry funded, Klein Funded, Blue Guardian Funded,
- Tools: TradingView, Metatrader5
- Indicators: Volume, Trading Sessions, EMA, RSI, Bollinger Bands, Smart Money Concept, SuperTrend, Footprints, Breakouts, Fibonacci, Orderbook, MACD

EDUCATION

Birla Institute of Technology & Science, Pilani — Mar 2026 – 2029

- Bachelor of Science in Computer Science