

TALHA RIAZ

+92 324 4324155 • talhariaz324@gmail.com • Pakistan • [LinkedIn](#) • [GitHub](#) • [Portfolio](#)

Full Stack Engineer

Full Stack Engineer building production products where payments, data volume, failure, and correctness are first-class engineering constraints. Experienced across 11-microservice payment systems, data-intensive platforms processing 5M+ events/day across 20+ exchange systems, multi-tenant applications supporting 12K concurrent sessions, and production AI workflows. Hands-on across product requirements, architecture, implementation, AWS infrastructure, and production recovery using event-driven systems, Kafka, PostgreSQL, Redis, and TypeScript.

Experience

WeiBlocks, Pakistan (Remote)

Jul 2024 – Present

Senior Full Stack Engineer

- Architected an 11-microservice payment platform, separating domain responsibilities and service data boundaries while using Kafka-based event-driven communication for asynchronous payment workflows. [Live](#) ↗
- Engineered payment workflows around correctness and reliable event delivery using Transactional Outbox, idempotency, and double-entry ledgering to safely handle duplicate and failed events across distributed money-moving operations.
- Built production recovery for failed Kafka delivery, persisting events that exhausted retries to a dedicated failed-events store with automated alerts and safe re-queue mechanisms, turning silent delivery failures into visible, recoverable incidents without manual intervention.
- Engineered CoinPerps to process 5M+ events/day across 20+ exchange systems and 6 news sources, isolating upstream integrations behind independent connectors with WebSocket/REST ingestion, normalization, reconnection, and failover. [Live](#) ↗
- Built 24 production AI pipelines and 8 specialized agents using RAG-style ingestion, model routing, event-driven scheduling, and health monitoring, reducing recurring manual work from ~6 hours/week to <15 minutes.
- Architected strict tenant isolation using PostgreSQL Row-Level Security and application-level tenant scoping, supporting 12K concurrent sessions with sub-300ms API latency while maintaining tenant boundaries across shared infrastructure.

Kadipay, Saudi Arabia (Remote)

May 2023 – Jun 2024

Full Stack Engineer

- Engineered product workflows across BNPL and subscription payments, working across client and backend layers to maintain consistent payment states through checkout, processing, failure, and reconciliation.
- Designed payment flows around idempotency, duplicate-event protection, webhook verification, reconciliation, and explicit failure states, preventing retries and asynchronous payment events from producing inconsistent application state.
- Used PostgreSQL as the durable source of truth for payment-affecting state while applying AWS SNS/SQS for reliable asynchronous side effects including notifications, dunning, and cache invalidation.
- Built and validated end-to-end payment workflows across success and failure paths, including duplicate checkout attempts, repeated webhooks, failed installments, missing callbacks, and stale client state.
- Implemented and integrated shared payment services across Consumer, Merchant, Admin/Ops, and Agent portals, keeping payment state and business rules consistent across multiple product surfaces.

Team4Tech Solutions, Canada (Remote)

Sep 2021 – Apr 2023

Full Stack Engineer

- Built and shipped a community mutual-aid platform from zero to production in 8 weeks as the sole technical hire, owning product architecture, financial data, payments, backend, frontend, and deployment.
- Designed the financial core around PostgreSQL ACID transactions and an immutable ledger, atomically committing contributions with ledger entries so balances remained reconstructable from transaction history.
- Engineered Stripe payment flows with idempotency keys, webhook verification, duplicate-event protection, reconciliation, dunning retries, and explicit failed-transfer states to prevent payment failures from becoming invisible financial inconsistencies.
- Reduced dashboard read latency by ~40% through query-profile-driven indexing and Redis cache-aside strategies with explicit invalidation for financial aggregates.

Core Competencies

Engineering: Event-Driven Systems • Microservices • Failure Recovery • Transactional Outbox • Idempotency • Saga • Double-Entry Ledger • Multi-Tenant Architecture

Technologies: TypeScript • Node.js • NestJS • PostgreSQL • Redis • Kafka • MongoDB • React • Next.js • React Native

Cloud & Infrastructure: AWS • Docker • Kubernetes • CI/CD

AI & Automation: Agent Orchestration • RAG • LLM Routing • Event-Driven AI Workflows

Education

University of Education, Bachelor of Science in Information Technology