# Tasnimul Mohammad Fahim

+880-193-490-8343 | faahim06@gmail.com | linkedin.com/in/insertfahim | github.com/insertfahim | MeFahim

## Summary

SRE-focused Software Engineer with hands-on experience building observable, fault-tolerant systems for multi-tenant applications handling 15K+ daily transactions. Specialized in cloud infrastructure, deployment automation, and incident management—delivering 10x latency improvements and 80%+ reductions in detection/recovery time. Experienced with Kubernetes, Prometheus, and SLO-driven reliability practices.

## Education

**BRAC University** Dhaka, Bangladesh
*Bachelor of Science in Computer Science and Engineering* *2021-2025*

## Technical Skills

**Cloud & Infrastructure**: AWS (EC2, S3, Lambda), GCP, Docker, Kubernetes, Terraform, Nginx, CI/CD Pipelines
**Observability & SRE**: Prometheus, Grafana, OpenTelemetry, SLO/SLI/Error Budgets, MTTR/MTTD, Incident Response
**Languages**: Python, Go, Java, TypeScript, JavaScript, SQL, Bash
**Databases & Storage**: PostgreSQL, MySQL, MongoDB, Redis, Schema Design, Query Optimization, Connection Pooling
**Frameworks**: FastAPI, React, Next.js, Node.js, Express.js, REST APIs, GraphQL, WebSockets, gRPC

## Professional Experience

**Software Engineer** Sep 2025 – Present
*The Flex & base360.ai (Sister Companies), United Kingdom* *Remote*

- Solved tenant data isolation for 500+ properties (15K+ transactions/day) by designing PostgreSQL Row-Level Security policies with PgBouncer connection pooling, eliminating cross-tenant data leaks and reducing auth overhead by 70%
- Diagnosed P95 latency spikes (800ms+) caused by N+1 queries; implemented Redis write-through cache with 15-min TTL and batch prefetching, dropping P95 to 85ms and cutting database load by 45%
- Built observability stack: structured JSON logging → Loki → Grafana dashboards with 12 SLI panels tracking error rates, latency percentiles, and saturation against defined SLOs; reduced MTTD from 45min to 8min, MTTR from 2hr to 25min
- Debugged WebSocket connection drops under load (TCP keepalive misconfiguration); redesigned with heartbeat mechanism, exponential backoff reconnection, and Supabase Realtime channels—achieved 99.7% message delivery
- Led toil reduction initiative: identified manual deployment as bottleneck (2hr+ per release); built GitHub Actions pipeline with parallel test execution, Docker layer caching, and blue-green deployment—now ship 3x daily with zero rollbacks

**Full Stack Developer** Jun 2023 – Feb 2024
*Doin Tech* *Remote*

- Inherited 3.2s page loads blocking 25K MAU; profiled with Chrome DevTools, implemented route-based code splitting, lazy-loaded below-fold components, and added Cloudflare CDN—achieved 95+ Lighthouse, 1.1s LCP, 99.8% availability
- Traced 22% cart abandonment to checkout timeouts; refactored synchronous payment flow to async with Redis job queue, added idempotency keys for retry safety—reduced timeouts by 80%, MTTR from 4hr to 30min, recovered $12K/month
- Discovered production bugs from untested edge cases; established Jest testing culture with 85% coverage mandate, pre-commit hooks, and CI gates—cut post-deploy hotfixes from 8/month to 2/month, improved uptime to 99.9%

## Key Projects

**The Flex PMS** | *React, FastAPI, PostgreSQL, Redis, Docker, Kubernetes*

- Designed micro-frontend architecture splitting monolith into 5 independently deployable modules behind Nginx ingress controller with path-based routing—enabled parallel team development, reduced deploy conflicts by 90%
- Profiled slow dashboard loads (6s+); optimized schema design with composite indexes on (tenant_id, created_at), query result pagination, and Redis cache warming on login—dropped P99 load time to 400ms for 3M+ monthly transactions
- Implemented defense-in-depth: OAuth 2.0 PKCE flow, JWT with 15-min expiry + refresh rotation, RBAC with 8 granular permissions, and immutable audit log (append-only table)—passed SOC 2 Type I readiness review

**Variant Analysis Evo2** | *Next.js, Python, Modal (Serverless GPU), FastAPI* GitHub

- Tackled GPU cost problem ($3.50/hr H100); built serverless inference on Modal with cold-start optimization (model weight caching), auto-scaling 0→50 replicas—cut per-inference cost from $0.12 to $0.05, 60% FinOps savings
- Integrated ClinVar/NCBI APIs with circuit breaker pattern and local fallback cache; maintained 99.5% data availability despite upstream rate limits, enabling 1K+ variant analyses/hour for genomics researchers

**IntelliMail** | *Next.js 14, tRPC, PostgreSQL, OpenAI, Stripe* GitHub

- Built RAG pipeline: chunked 100K+ emails → generated embeddings via OpenAI ada-002 → stored in pgvector with HNSW index—achieved 50ms semantic search vs 2s+ keyword search, 40x speedup
- Solved Stripe webhook reliability: implemented idempotent event processing with Redis deduplication, dead-letter queue for failed events, and automated retry with exponential backoff—reached 98% payment success, $8K MRR