

MUHAMMAD TAQI HAIDER

+92-330-287-7064 | taqihaid591@gmail.com | github.com/Muhammad-Taqi-Dev | linkedin.com/in/taqih1 | Portfolio

PROFESSIONAL SUMMARY

Backend Engineer specializing in performance optimization and distributed systems (NestJS, Next.js, PostgreSQL, Redis). Cut API latency 63% and database load 60% across two production systems, and independently architected an ML-powered finance platform hitting 88% categorization accuracy. Promoted to Associate Software Engineer in 3 months at QBS Co. – currently owning authentication and monitoring infrastructure for a live SaaS platform.

WORK EXPERIENCE

QBS Co.

Associate Software Engineer

Feb 2026 – Present

- Cut API response latency from 1.6s to under 600ms by introducing Redis caching for third-party responses, eliminating redundant external calls on high-traffic monitoring screens.
- Engineered SEO-optimized SSR dashboards (Next.js/TypeScript) surfacing 21 real-time KPIs across org, individual, and product-level monitoring, with live-stream log playback via AWS S3.
- Implemented authentication and RBAC across 5+ microservices, including JWT authentication, refresh token handling, Redis-backed session management, and role-based authorization.

Trainee Software Engineer

Nov 2025 – Jan 2026

- Replaced a manual admin workflow by building an RBAC/IAM interface in Next.js managing 278 permissions across workspace-level feature gates with bulk assignment support.
- Built two production backend services – inventory management and SLA workflow orchestration – using gRPC for real-time inter-service communication across Dockerized microservices.
- Integrated the WhatsApp Business API to automate live ticket delivery and customer notifications.

KoderLabs

Associate Software Engineer

Mar 2025 – Oct 2025

- Reduced per-request DB hits from 6–8 queries to 2–3 by refactoring .NET Core service logic under Clean Architecture, preserving full API backward compatibility.
- Eliminated a document generation bottleneck by architecting an event-driven async PDF pipeline (Azure Functions, Azure Queue Storage), cutting processing time from ~1.6s to 400–500ms per document; separately implemented SignalR for real-time chat notifications across the platform.
- Implemented polygon-based geospatial filtering for real estate listings using PostGIS and Google Maps API, optimizing spatial queries to reduce execution time from 2.1s to 600ms on a multi-tenant SaaS platform.

PROJECTS

TrackWise: AI-Powered Personal Finance Manager

May 2025 – Present

Final Year Project (team of 3) — backend & ML lead. AI-driven finance platform; live on Azure (backend/DB) and Vercel (frontend) with active beta users.

- Architected a modular-monolith NestJS/PostgreSQL backend with 56 REST endpoints across income & expense tracking, savings goals, budget planning, and category-wise summaries; deployed on Azure, used by 10 beta testers on real transactional data.
- Built a Flask ML microservice (3 REST endpoints) on Hugging Face Spaces, called internally by the main backend, hosting transaction categorisation (SBERT + Logistic Regression, 88% test accuracy on held-out data), expense forecasting (Facebook Prophet, 3+ months history), and real-time anomaly detection (AUC-ROC: 0.87); mitigated cold-start latency via caching and rate limiting.
- Wrote Jest unit/integration tests for core services; set up GitHub Actions CI/CD for automated testing and deployment; implemented email-based 2FA and account verification.

EDUCATION

NED University of Engineering & Technology

Oct 2022 – June 2026

Bachelor of Science in Computer Science and Information Technology (Karachi, Pakistan)

TECHNICAL SKILLS

Languages & Frameworks: TypeScript, JavaScript, Python, C# — NestJS, Next.js, React, Flask, .NET Core

Architecture: REST APIs, gRPC, Microservices, Clean Architecture, Event-Driven Architecture

Databases & Caching: PostgreSQL, PostGIS, Redis

Cloud & DevOps: AWS S3, Azure Functions, Azure Queue Storage, SignalR, Docker, CI/CD, Git

Security & Testing: JWT, OAuth2, RBAC, 2FA, Jest, Postman

Other: 240+ problems on [LeetCode](#) & [HackerRank](#)