

ALI FAROOQ

Backend Engineer · AI Workflow Automation and Integrations

Karachi, Pakistan | hello@alifarooq.dev | +92 316 7902206 | linkedin.com/in/itsalifarooq | alifarooq.dev

SUMMARY

Backend engineer with two years building AI-automation systems in NestJS and TypeScript, designed for the failure case first: tokens that expire, integrations that go quiet, payments that must not capture twice. Takes features end to end — PostgreSQL schema, BullMQ background jobs, third-party integrations (Gmail, Vapi, Stripe, AWS), and the Next.js dashboard on top.

EXPERIENCE

Software Engineer

Zenkoders

Oct 2024 - Present

Karachi, Pakistan

Delivered AI-automation and backend systems across multiple client products in healthcare, fintech, and SaaS. Won the company's **Excellence in Work** award for delivery.

A healthcare services provider · AI automation suite

- Built an AI voice-calling and booking platform linking Pipedrive CRM to Vapi voice agents through BullMQ queues. It processes **7,000+ inbound leads a month** and books appointments using holiday and leave-aware Google Calendar scheduling. (*NestJS, Next.js, TanStack Query, PostgreSQL/Prisma, AWS RDS*)
- Built a Gmail automation and AI email-drafting system for **100+ consultants** handling **40 to 50 client messages a day**, with OAuth token and watch renewal, full-sync recovery queues, and retrieval-augmented (RAG) drafting verified through Tavily for compliance-safe replies. (*NestJS, Drizzle, BullMQ, Gmail API, OpenAI*)
- Designed document-extraction pipelines that turn unstructured PDFs and images into validated, schema-typed data using vision LLMs and confidence-aware section reduction, cutting manual form-filling substantially across intake and R&D tax-incentive workflows, and automated the financial reporting into formatted Google Sheets. (*AWS S3/SQS/Lambda, OpenAI, Zod*)

A virtual-phone-number & SMS platform, a restaurant platform and a community platform

- Built a virtual-number and SMS SaaS with a wallet and ledger system (hold, capture, release, plus overdraft protection), secured by a four-layer RLS and JWT-claim access model over a third-party provider API. (*Supabase, Drizzle, Next.js 16*)
- Built backend platforms using CQRS, RBAC, and i18n, plus real-time chat and notification features (Socket.IO, unread-count APIs), documented with Swagger and Scalar. (*NestJS, Prisma, TypeORM, Redis*)

Across all products · reliability & delivery

- Improved reliability with structured logging, idempotent queue processors, test suites for schedulers, auth, and sessions, and ORM migrations; automated GitHub Actions deployments to EC2 and expanded CI/CD across several client codebases.

PROJECTS

A property-management SaaS

Lead full-stack developer

Multi-tenant platform for property managers to onboard residents, dispatch maintenance tickets to technicians, and handle staff roles and billing. Lead developer on a five-engineer team: built Stripe subscription billing, real-time ticket dispatch with live accept and decline updates, and CSV bulk-import wizards for residents and properties modeled as reducer state machines, on a queue-backed backend.

Tech: Turborepo, NestJS, BullMQ, Redis, Next.js 16, React 19, Stripe, Supabase, PostgreSQL

A multi-tenant workshop-management SaaS

Solo project

SaaS that lets automotive workshops manage jobs, estimates, invoices, inventory, and staff across multiple sites. Solo-built the entire backend: a tiered RBAC permission engine, about 22 domain modules, Drizzle ORM with a Unit-of-Work transaction layer, and CI/CD to a self-hosted VPS. Designed a Backend-for-Frontend cookie proxy that keeps cross-origin auth working under Chrome's third-party-cookie deprecation.

Tech: NestJS 11, PostgreSQL, Drizzle, AWS S3, Next.js 16, TanStack Query, Zod

Trademark Register System · Legal Services Firm

Solo project

Desktop application replacing a Microsoft Access system for a legal services firm, with a web port in progress. Parses government trademark reports published as Word documents into a shared PostgreSQL register — an upstream format with no API, no schema, and no notice when it changes. Desktop client and web portals read the same register.

Tech: C#, .NET, ASP.NET Core, Next.js, PostgreSQL

result-kit, result-kit-lint & drizzle-tx · Open-Source TypeScript Libraries

Published on npm

One conviction applied twice: model failure as a value instead of throwing through the service layer. **result-kit** provides the Result type and a fluent async pipeline whose generic inference holds across deep chains; **result-kit-lint** makes the compiler enforce it, failing the build when a result goes unchecked. **drizzle-tx** applies the same idea where the ORM fights back — Spring-style implicit transaction propagation for Drizzle on Node's AsyncLocalStorage, inverting Result back to a throw at exactly one rollback boundary.

github.com/alifaroo-q/result-kit · github.com/alifaroo-q/result-kit-lint · github.com/alifaroo-q/drizzle-tx

Tech: TypeScript, Drizzle ORM, AsyncLocalStorage, ESLint, Vitest, Changesets

TECHNICAL SKILLS

Languages	TypeScript, JavaScript, Python, SQL, Java, C#, HTML, CSS
Backend	NestJS, Node.js, Hono, Express, REST APIs, CQRS, Microservices, BullMQ, WebSockets, Socket.IO
Frontend	React 19, Next.js 16, TanStack Query and Table, Zustand, React Hook Form, Zod, Tailwind CSS, shadcn/ui, Radix
Data and Storage	PostgreSQL, Prisma, Drizzle, TypeORM, Redis, Supabase (RLS)
AI and Automation	OpenAI, Vapi, Vercel AI SDK, LangChain, RAG, Vision and Document Extraction, Prompt Engineering, Claude Code, Cursor
Cloud and DevOps	AWS (S3, SES, SQS, Lambda, EC2, RDS, IAM, CloudWatch), Docker, GitHub Actions, CI/CD, Turborepo
Practices	Testing (Unit tests with Vitest, E2E tests with test Database), JWT and OAuth, RBAC, Swagger and Scalar, Structured Logging, Agile, Git

EDUCATION

DHA Suffa University

Oct 2020 - Jul 2024 · Karachi, Pakistan

B.S. in Computer Software Engineering

Gold Medalist, graduated top of batch