

Muhammad Arab

Senior Backend Engineer · System Design, Payments and Order Processing

Pune, India · Open to remote and relocation · muhammadarab@gmail.com · +91 70418 81109 · muhammadarab.com

Backend engineer who designs the business logic companies run on, from order processing and payments to the rules that decide who can see which data. Nearly four years across Node.js, PostgreSQL, and NetSuite, the last three at a consultancy delivering for two clients in parallel, usually from a loose brief to production against a fixed deadline. I make the architectural calls on systems that are already live, where every change has to land without disrupting the business.

PROFESSIONAL EXPERIENCE

Software Developer · Bee Logical Software

Sep 2023 - Present · Remote (clients in the US and EU)

Two clients in parallel since May 2026. Awarded Innovation Excellence 2025 and Collaborative Spirit 2024.

Saralux · Builds MEM360, an association management system on NetSuite

Sep 2023 - Present

- Designed and led the build of the order processing engine that now handles every transaction in MEM360, a product six associations and nonprofits each run in their own NetSuite environment. In the old flow, one failure stopped the whole order. The new engine runs each order as a sequence of tracked steps, from registration to billing, so a failed step can be retried on its own and failed orders can be reprocessed in bulk.
- Re-architected one association's order lifecycle, moving its payment-status and billing logic out of NetSuite workflows into a state machine in code that sets each order's status and billing stage, including renewals. Workflows and scripts had been updating the same records concurrently. The single deterministic flow eliminated those race conditions and made order processing idempotent, and I shipped it as an installable SuiteApp.
- Found an IDOR flaw that let a logged-in user see other customers' saved cards and addresses by changing one ID in a request. The real problem was the server trusting identity sent by the browser, so I fixed it at that level. Across 69 files in six modules the server now takes the customer from the login session and checks its own execution context, and I rewrote guest checkout so guest identity is also established on the server instead of trusted from the client.

Dog go · Dog-care marketplace connecting owners with providers in the EU

May 2026 - Present

- Reworked provider payouts through Stripe Connect in the Node.js API. Bookings only checked that a provider had a Stripe account, so an owner could be charged for a provider who couldn't receive payouts yet, leaving the money stuck. Booking now requires an account that can both take charges and pay out, and every provider is on automatic daily payouts. I also dropped an unneeded Stripe capability whose identity checks were delaying providers' first payouts.
- Moved booking validation ahead of payment. Owners used to hit a full slot or unavailable provider only at the last step. I put the booking rules into one shared check that the app calls before payment and creation runs again, so both always agree and a booking that became invalid in between is still rejected.

INDEPENDENT PROJECTS

Masjidly · Mosque directory built and run solo on Next.js and PostgreSQL

Feb 2024 - Present

- Built the pipeline that imported 27,714 mosques across India from OpenStreetMap and Google Places. Since the same mosque often appears in both, records pass a matching cascade, exact source ID first and then a similar name within 100 meters, that caught 623 duplicates. Low-confidence records go to a review queue instead of going live.
- Kept search in PostgreSQL instead of the app. One stored function handles up to nine optional filters, including fuzzy name matching and distance, in a single query, and triggers keep derived columns correct on every write.
- Built Google sign-in with account linking that refuses to merge two people's accounts when an identity already belongs to someone else, and role-based access so mosque admins are verified before they can manage a listing.

Agentic Delivery Pipeline · Spec-driven development with AI coding agents

2026

- Built a five-stage set of Claude Code skills that takes a project from a written spec to a reviewed app. Planning writes the spec and a task-by-task plan, two build stages implement the data layer and interface, review checks every acceptance criterion without touching code, and a retro feeds lessons back into the skills. Used it to build FinanceTracker.

TECHNICAL SKILLS

Languages and runtime: TypeScript, JavaScript, SQL, Node.js 20, Express, Zod, Drizzle ORM, OAuth 2.0/JWT, Vitest, Supertest

Databases: PostgreSQL (PostGIS, pg_trgm, Row-Level Security, Stored functions, Triggers), MySQL, Supabase

Architecture: System design, REST APIs, State machines, Idempotency, Pub/sub, Batch jobs, Feature flags, RBAC

Platforms: Stripe Connect, NetSuite SuiteScript (User Event, Map/Reduce, RESTlets), SuiteApps, Google Cloud, Azure DevOps

AI-Assisted Delivery: Claude Code, Spec-driven development, Agentic coding workflows

EDUCATION

MIT World Peace University, Pune · B.Tech, Computer Science and Engineering (CGPA 9.02/10)