

Gazi Shahi

gazishahi@gmail.com | github.com/gazishahi | linkedin.com/in/gazishahi | gazishahi.com

TECHNICAL SKILLS

Languages: JavaScript, TypeScript, Python, C++, Swift, Kotlin, SQL (PostgreSQL), Svelte, HTML, CSS
Frameworks: React, React Native, Node.js, Next.js, SvelteKit
Databases: PostgreSQL, DynamoDB, Neo4j
Infrastructure: AWS, Docker, Supabase, Firebase, Stripe API

EXPERIENCE

Muslim American Society of NY

New York, NY

Software Engineer

Jun. 2026 - Present

- Built and shipped a multi-tenant intranet for MAS-NY headquarters, its sub-chapters, and affiliate partner centers (Next.js 16, React 19, PostgreSQL/Prisma, Better Auth); 30+ data models across 15+ modules including ticketing, announcements, events with a Hijri calendar, org chart, and grant-packet checklists.
- Designed the tenancy and access model: sub-chapters roll up to a headquarters tenant, partner centers get a restricted portal; built on Better Auth's organization plugin with role-based permissions, Zod-validated Server Actions, and audit logging.
- Led the Reverts Reconnect website rebuild (TanStack Start, Sanity CMS, Cloudflare Workers) for a 3,000+ member community — bilingual EN/ES, CMS-editable by non-technical staff, 128 automated tests, WCAG 2.1 AA audited.

Khidma Inc

New York, NY

Software Engineer (Founding Engineer)

Aug. 2025 - Present

- Led architecture and full-stack development of a multi-tenant event management platform serving 4+ organizations and processing **\$30,000+** in ticket transactions using React, TypeScript, Supabase, and Stripe
- Designed normalized PostgreSQL schema with strict foreign key constraints and row-level security (RLS) policies to guarantee tenant isolation and prevent cross-organization data leakage across **2,000+** records
- Implemented idempotent ticket purchasing workflow using transactional writes and server-side validation to eliminate duplicate charges during concurrent purchase attempts
- Architected hierarchical role-based access control (RBAC) model supporting admins, volunteers, and attendees with enforced permission boundaries at query level
- Utilized vision language models to swiftly process event flyers and pre-fill event information, improving event creation speed by over 3 times
- Implemented event lifecycle workflow to enforce valid status transitions and prevent invalid event state updates

Relay

New York, NY

Software Engineer

Feb. 2024 - Aug. 2025

- Developed cross-platform mobile applications for iOS (SwiftUI) and Android (Kotlin) used by **1,500+** users, integrating REST APIs for authentication, content delivery, and progress tracking
- Reduced application memory usage by **30%** by removing unnecessary dependencies and resolving retain cycles that caused objects to persist longer than needed
- Improved API request latency by **40%** by restructuring network calls and preventing redundant fetches triggered by UI state updates
- Conducted TestFlight beta testing with early users to identify performance issues and UI bottlenecks prior to production release
- Delivered production builds to the **App Store and Google Play within 4 months**, presenting the application to 500+ attendees at conferences and campus events

Research Foundation of CUNY

New York, NY

Blockchain Research Intern

April 2022 - April 2023

- Developed an encrypted health records (EHR) sharing system in a team of 3 that improves synchronization of data between healthcare facilities within 3 months, allowing users privacy and ownership of their data
- Deployed a decentralized framework by utilizing Ethereum blockchain smart contracts with Inter-Planetary File System (IPFS) and NuCypher encryption protocol in 2 months, appropriately hashing transactions to ensure user privacy is maintained
- Established HAPI FHIR server via Java to automatically parse patient data, ensuring compliance with critical patient privacy regulations and providing consistent structure and interoperability among external healthcare entities

PROJECTS

Side - Mac Native ADE

Present

- Built a native Mac development environment from scratch, organized around git-backed "tracks": durable units of work with intent, agent workstreams, and a reviewable path to the main branch instead of permanent sidebars and panels. Multi-window with shared per-project state and automatic stash/pop across branch switches.
- Designed a provider-agnostic agent runtime: SSE-streamed tool-use loop behind a single ChatModelProvider protocol, where write tools emit inspectable diffs that never touch disk without human approval. Owned outside the view layer, so a run survives stage, track, and window teardown mid-approval.
- Implemented an LSP client from scratch: Content-Length-framed JSON-RPC over stdio, per-binary process lifecycle, 7 servers across 6 languages, powering inline diagnostics, autocomplete, and go-to-definition.

EDUCATION

The City College of New York

New York, NY

B.E. Computer Engineering