

Mohammed Faizan Ahmed

info2faizanahmed@gmail.com | [+91-9391281076](tel:+91-9391281076) | github.com/AhmedFaizanDev | linkedin.com/in/mohdfaizanahmed | faizanahmed.in

Summary

Backend and systems engineer (B.Tech Computer Science, 2027). Built an embeddable SQL incremental view maintenance engine benchmarked at 37K updates/sec, p99 43 μ s, verified by 570 differential tests against an independent recompute oracle; plus a multi-tenant SaaS backend with isolation enforced in middleware. Two years of production experience across Django, TypeScript/Node, PostgreSQL, Redis and Docker.

Education

Mahindra University (Mahindra École Centrale)
B.Tech, Computer Science and Engineering

2023 – 2027
Hyderabad, India

Technical Skills

Languages: Python, TypeScript, JavaScript, SQL, C/C++ (coursework)

Backend & APIs: Django, Django REST Framework, Express.js, Node.js, REST API design, JWT authentication, OAuth 2.0 (PKCE), RBAC, OpenAPI

Databases & Data: PostgreSQL, SQLite, Redis, schema design, indexing, query optimization, incremental materialization, change data capture

Infrastructure & DevOps: Docker, Docker Compose, AWS EC2, Nginx, GitHub Actions CI/CD, Linux/Unix, shell scripting, Ansible

Experience

Artificial Intelligence Intern | *AI Linc Technologies Pvt. Ltd.*

Jul. 2026 – Aug. 2026

- Built backend services powering generative-AI product features for client work, including automated video content generation and long-form document generation pipelines, exposed as authenticated REST endpoints with request validation, schema-enforced structured outputs, retry with exponential backoff and per-provider timeout handling.
- Cut median latency on the generation path by 38% by replacing synchronous provider fan-out with a queued worker model, moving multi-step generation off the request thread.
- Reduced repeat inference calls roughly 30% by caching deterministic intermediate results in Redis, keyed on normalized request content so semantically identical requests reuse prior work.

Full Stack Engineer | *AI Linc Technologies Pvt. Ltd.*

Jan. 2026 – Jun. 2026

- Owned 20+ backend features end to end: data model and migrations, serializer and permission layers, REST endpoint design and the contracts consumed by the frontend team.
- Integrated ML models into production backend services behind versioned REST contracts, standardizing request/response schemas and adding input validation so model changes did not break API consumers.
- Raised automated test coverage on owned backend modules from 22% to 68%, targeting the serializer and permission logic where regressions had previously reached release; reviewed 40+ teammate pull requests in agile delivery cycles.

Full Stack Developer (Freelance) | *Catalyst Connect, Hyderabad*

Feb. 2026 – Mar. 2026

- Built the backend for a production exhibition and stall management platform delivered to a paying client, using Node.js, Express and PostgreSQL, with JWT authentication, role-based access and stall booking that prevents conflicting concurrent reservations.
- Shipped on Docker with PostgreSQL and Redis behind a CI/CD pipeline; used to run 3 exhibitions covering 200+ stalls and their payment records.

Backend Developer | *YuVi Collab Technology Pvt. Ltd.*

Jun. 2024 – Sep. 2024

- Built and maintained 25+ Django REST API endpoints, authentication, and background workflows over PostgreSQL for client-facing systems.
- Cut p95 latency on the heaviest list endpoint from 1.4s to 310ms by eliminating N+1 query patterns with `select_related/prefetch_related` and adding composite indexes matched to the actual query predicates.

Head of Tech & Design | *Qubit Club, Mahindra University*

Jul. 2025 – Jun. 2026

- Led a 6-person technical team, owning the club's official website and the technical delivery for 8 workshops and events reaching 400+ attendees.

Projects

ivm: Embeddable Incremental View Maintenance Engine | *Python, SQL, SQLite, pytest, GitHub Actions*

2026

- Built a library that keeps a declared SQL view correct as its base tables mutate by applying only the deltas instead of recomputing; this is the capability streaming databases provide as a separate distributed service, packaged here as an importable dependency. Implements the DBSP model, in which every row carries a signed weight (insert = +1, delete = -1).
- Implemented incremental operators for projection, filter, INNER/LEFT/RIGHT/FULL joins, and GROUP BY with COUNT/SUM/AVG/MIN/MAX, handling the retraction cases that break naive implementations: recovering the next MIN/MAX after the current extreme is deleted, and flipping outer-join rows between matched and NULL-padded states as matches appear and disappear.
- Designed the test strategy from the failure mode: incremental maintenance fails *silently*, since a wrong aggregate after a delete still looks like a valid number. Built an independent recompute oracle sharing no code with the incremental path. **570 tests** assert `incremental == oracle` after every delta across randomized insert/delete streams drained to empty, with the SQL front-end additionally cross-validated against real SQLite. CI on Python 3.10–3.12.

- Benchmarked **37,000 single-row updates/sec** at **p50 21 μ s** / **p99 43 μ s** on a 50,000-row base: $\approx 3,600\times$ faster than full recompute for a small GROUP BY update and $\approx 33\times$ even when 2% of the base changes, with maintained state of ≈ 0.2 bytes/row for aggregates.

Blackgrid: Multi-Tenant Inventory SaaS Platform | *Django 5, DRF, PostgreSQL, Redis, Docker, Nginx* 2025

- Built a multi-tenant backend serving multiple independent businesses from one deployment, with isolation enforced centrally in middleware rather than left to per-view developer discipline: tenant IDs are signed into the JWT and validated against every tenant-scoped route, and the **X-Tenant-ID** header is stripped on cross-origin requests to block forgery.
- Shipped 40+ REST endpoints with Owner/Admin/Viewer RBAC and drf-spectacular OpenAPI docs, covered by 84 backend tests including cross-tenant access attempts that must return 403, so isolation is verified rather than assumed. Deployed on Docker Compose with PostgreSQL 15, Redis 7 and Nginx.

Fault-Tolerant LLM Content Generation Pipeline | *TypeScript, Node.js, Express, Docker, AWS EC2* 2025

- Built a staged generation pipeline producing ≈ 250 -page structured documents from a single topic through **≈ 186 sequenced LLM calls**, orchestrated behind an Express API with a batch CLI driven from CSV.
- Made long, expensive runs survivable: checkpoint/resume at unit granularity plus retry rounds mean a failure at call 150 resumes without regenerating completed work, and object-store state is used as the idempotency key so re-running a batch skips already-produced outputs instead of duplicating them.
- Held marginal cost to **$\approx \$0.12$ per document** through model tiering and micro-summary context chaining instead of full-history prompts; containerized with Docker Compose and deployed on AWS EC2.

PyTor: Peer-to-Peer BitTorrent Client | *Python, TCP Sockets, Multithreading* 2024

- Implemented a P2P file-sharing client against the BitTorrent specification using raw TCP sockets: bencode parsing of **.torrent** metadata, tracker announce, peer handshake, and length-prefixed message framing over the TCP byte stream, without relying on an existing BitTorrent library.
- Verified each downloaded piece against its SHA-1 hash before commit and re-requested corrupt pieces from alternate peers, sustaining 4.2 MB/s across 30 concurrent peers under a thread-per-peer model with a shared piece-assignment queue.

Achievements

Winner: SUDHEE 2025 Hackathon | *CBIT, Hyderabad* Mar. 2025

- Built a real-time speech-to-Indian-Sign-Language translator, placing first among 60+ competing teams; owned the streaming speech-to-text pipeline and gesture rendering layer, tuning it to hold end-to-end latency under 1.2s so output tracked live speech.

3rd Place: AminoVerse Health-Tech Hackathon | *Mahindra University* Apr. 2025

- Built the backend data-integration layer and Qdrant-backed vector retrieval over four bioinformatics databases (PDB, UniProt, StringDB, DrugBank) for a gene-disease analysis assistant, delivered in a 48-hour build.

Certifications

Red Hat Linux Administration III: Automation with Ansible (Udemy, Feb. 2025) | Linux system administration, shell scripting, Ansible automation.

Oracle Cloud Infrastructure 2025 Certified Generative AI Professional (Oct. 2025) | LLMs, RAG pipelines, semantic search, vector databases.