PORTFOLIO AND WORK RECORD · NOVEMBER 2026

Kartik Jain

Security & Backend Engineer

Mumbai · career@jkartik.in · jkartik.in
github.com/KartikJain14 · linkedin.com/in/kartikjain1410

This document is the long-form record behind the one-page résumé. Each role and project is described with its context, the approach taken and the outcome. Ownership is stated plainly: solo, lead, co-lead, maintainer or contributor.

CONTENTS

01 Experience 3
02 Security research 5
03 Selected projects 6
04 Leadership and community 11
05 Achievements 12
06 Skills, with evidence 13
07 Education 14
08 Links 15

At a glance

Kartik Jain is a security and backend engineer in Mumbai and a B.Tech Information Technology student at NMIMS MPSTME (2024–2028, CGPA 8.34/10.0). As a Software Engineer Intern at VisionRise Technosoft, Kartik built a white-label learning platform end to end and shipped it as two Google Play apps. As a Security Engineer Intern at R.U.D.R.A Cybersecurity, a Mumbai VAPT firm, Kartik found two high-severity flaws in a banking client's portal and automated attack-surface monitoring. Alongside both: a $1,500 bug bounty from Meta, vulnerability reports to Microsoft and (through CERT-In) the Mumbai Police, freelance backend and incident-response work, and the technical side of several college organisations, most recently as Chairperson of ACM MPSTME.

STRONGEST AREAS

Application security
Web-application penetration testing, manual code review and coordinated disclosure, with findings reported to Meta, Microsoft (MSRC) and CERT-In. LLM red teaming: 1st place on the global leaderboard of chat.win's AI Jailbreakathon.

Backend engineering
APIs in TypeScript/Node.js and Python on PostgreSQL, with the decisions that matter enforced on the server: authentication, role-based access, signed payment webhooks, encrypted media delivery and race-free check-ins.

Running production
Hardened Ubuntu servers, Docker and Coolify CI/CD, automated backups to cloud storage, self-hosted mail with SPF, DKIM and DMARC, and incident response when a client platform was hit by ransomware.

IN NUMBERS

$1,500
bug bounty awarded by Meta for a privilege escalation in WhatsApp

276
REST routes across 20 modules in the LMS backend built at VisionRise

~4,000
students on the official Railway Concession system Kartik maintains

16,000+
users on a client platform Kartik hardened and brought back after ransomware

~$20,000/yr
saved by automating attack-surface monitoring at R.U.D.R.A

600+ / 500+
delegates served by the Delego and conc-qr event backends

1,100+
hackathon participants onboarded automatically by a Discord bot

~500
learners in Programming For Everyone, a record, as ACM Chairperson

1st
place on the global leaderboard of chat.win's AI Jailbreakathon

LOOKING FOR

Internships in security engineering or backend development, and freelance engagements in web-application security testing and backend development. Graduating in April 2028. Contact: career@jkartik.in.

01

Experience

VisionRise Technosoft Pvt. Ltd.

Apr 2026 – Oct 2026 · Mumbai

Software Engineer Intern. Built the company's white-label learning platform and ran its production server.

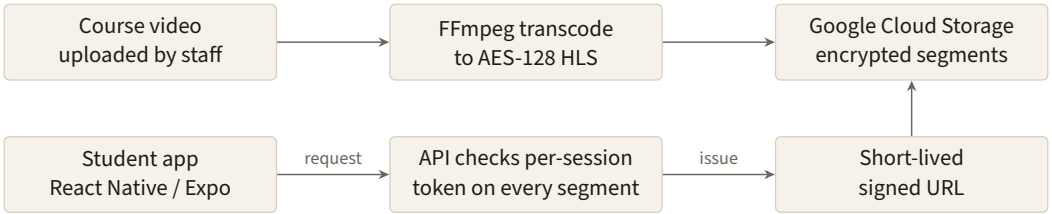
COMPANY	Edtech and software company behind the Rise Academy / Rise Institute and eGuru brands (risetechnosoft.com)
OWNERSHIP	Created and wrote nearly all of the LMS v2 codebase and the eGuru fork; contributed to the earlier, team-built LMS v1
STACK	TypeScript, Node.js, Express, PostgreSQL, Drizzle ORM, Zod, JWT, React Native, Expo, FFmpeg, Google Cloud Storage, Gemini, Zoom, Docker, Coolify, Ubuntu 24.04 LTS
LIVE	Rise Academy and eGuru on Google Play · riseinstitute.tech · eguru.app

Context. LMS v2 is a white-label learning platform: one backend and one app codebase that ship under more than one brand, covering courses, live classes, assessments and paid video for students on Android, iOS and the web. Kartik built it from the database schema to the store listing, then forked it into eGuru, a second branded app.

Backend architecture. A Node.js/Express API in TypeScript (~40,000 lines), organised into 20 domain modules that together expose 276 REST routes: admin, AI, analytics, assignments, audit, auth, calendar, certificates, courses, enrollments, lessons, live classes, media, modules, notifications, payments, quizzes, reports, settings and users. Inputs are validated with Zod at the boundary, data lives in PostgreSQL with schema changes tracked as Drizzle migrations, and the repository carries 25 test files.

One app, three platforms. The client is a single React Native/Expo codebase (~45,000 lines) that builds for Android, iOS and the web, with push notifications and over-the-air updates through Expo, so fixes reach installed apps without waiting on a store release.

Protecting paid content. Paid video is the asset most worth protecting. Uploads go through an FFmpeg transcoding pipeline that produces AES-128-encrypted HLS. Playback needs a per-session token that the API validates on every segment request, and media is fetched through short-lived signed Google Cloud Storage URLs, so a copied link expires quickly and does not work outside the session that requested it. PDFs stream inside the app with no download surface, and course certificates can be verified through a SHA-256 hash.



Paid-video path: content is encrypted when it is transcoded and is reachable only with a session token and an expiring URL.

Authentication and access. Students sign in with Google or with a PIN tied to a phone number verified over WhatsApp. Sessions use JWT access and refresh tokens, role-based access separates admins, instructors and students, and admin actions are written to an audit log.

Features and integrations. Kartik built Live Classes v1 on Zoom: classes are orchestrated from the platform, Zoom's webhooks are signature-verified before they are trusted, attendance and recordings are captured,

and Gemini then generates a quiz from the class content. Around this core Kartik shipped courses, quizzes and assignments, UPI payment verification, push and email notifications, and bulk enrollment.

Production server. Kartik provisioned and hardened the company's primary production server on Ubuntu 24.04 LTS with a standard baseline: SSH key-only access, a firewall, unattended security upgrades and fail2ban. The apps and PostgreSQL run behind Coolify, which handles CI/CD and deployments; container and application logs are collected through Coolify and Docker; and automated backups go to Google Cloud Storage.

IT administration. Kartik administered the company's GitHub organisation (15 repositories, about 5–10 members) and its Google Workspace directory of 35+ accounts, covering accounts, access and permissions.

Other work. Contributed to the earlier, team-built LMS v1: CI/CD, timezone-safe class scheduling with the PostgreSQL session pinned to IST, course-invite and access-request flows, and Zoom fixes. Built a government procurement-management prototype (FastAPI and React) for a client demo, and the eguru.app marketing site as a Next.js static export.

Outcome. Two apps live on Google Play, Rise Academy and eGuru, built from one codebase and one backend, and deployed through a pipeline onto a server that Kartik set up and maintained.

R.U.D.R.A Cybersecurity Pvt. Ltd.

May 2025 – Jul 2025 · Mumbai

Security Engineer Intern for three months at a Mumbai-based VAPT and offensive-security firm: client testing, internal tooling and purple-team exercises, using BBOT, Semgrep, Neo4j, Nuclei and Sliver C2.

Attack-surface monitoring. Kartik automated recurring external attack-surface monitoring with BBOT, cutting manual reconnaissance and saving ~$20,000 a year.

Static analysis at scale. Kartik single-handedly built a custom Semgrep (SAST) module with Neo4j-based graph visualisation, for analysing large codebases.

Banking client. Manual code review of a banking client's web portal found two high-severity vulnerabilities: a local file inclusion, and broken access control that exposed restricted functionality to unauthorised users. Both were reported to the client, and Kartik then automated their detection with custom Nuclei templates.

Purple teaming. Deployed and operated the Sliver C2 framework for purple-team exercises that emulated real-world intrusions.

Outcome. Two high-severity findings for a client, a recurring recon task automated with a ~$20,000-a-year saving, and an in-house SAST module.

Freelance

May 2025 – Oct 2026

Self-employed Backend & Security Engineer. Two separate engagements, not continuous; clients are not named.

Incident response and hardening

May – Jun 2025 · about one month

A freelance client ran a web platform with 16,000+ users. Kartik hardened its server infrastructure, and when the platform was hit by ransomware, led the incident response: contained the attack and restored service through rapid backup and recovery.

sports-live: tournament auction and scoring platform

Aug 2026 – Oct 2026

Built for a client and used at ELEV8, a pickleball tournament in which ~200 players were auctioned into 8 teams. The platform covers player registration and verification, a purse-based live player auction with a board for the projector, fixtures, live scoring and standings. Venue Wi-Fi is unreliable, so writes go into an offline queue in IndexedDB and are sent when the connection returns.

OWNERSHIP	Solo
STACK	Next.js 16, Supabase (PostgreSQL with row-level security, Realtime, Storage), server-side image processing, IndexedDB, Vitest

02

Security research

Kartik reports vulnerabilities to the affected organisation or through its disclosure program, and the technical detail below goes no further than what is published at jkartik.in/research. The work starts by hand, with manual code review, Burp Suite and Nmap; what is learned is then automated with Nuclei, Semgrep and BBOT.

ORGANISATION	FINDING	CHANNEL AND STATUS
WhatsApp (Meta)	Privilege-escalation vulnerability in WhatsApp.	Meta bug bounty program; $1,500 bounty awarded (2022)
Microsoft	Access-control and information-disclosure issues in Intune, Entra/Azure and the Exchange Admin Center. Organisation member data was served before the authorization check completed, a time-of-check/time-of-use gap spanning Intune, Entra and Azure. Standard users could bypass “restricted” policies through the Exchange admin portal to edit directory name fields and join closed groups.	Reported to the Microsoft Security Response Center (MSRC)
Mumbai Police	Stored cross-site scripting on the official website.	Disclosed responsibly, in coordination with CERT-In
The Souled Store	Exposed staging infrastructure that could have allowed a full system compromise.	Reported
Belgian Waffle Co.	Source-code and credential leak that opened the door to a potential CRM takeover.	Reported
Banking client	Local file inclusion and broken access control, both high severity, found through manual code review; detection then automated with custom Nuclei templates (p. 4).	Reported to the client during a R.U.D.R.A engagement

DEFENSIVE WORK

Ransomware response and infrastructure hardening

Freelance · May – Jun 2025

Hardened the server infrastructure of a web platform with 16,000+ users and, when ransomware hit it, led the response that contained the attack and restored service through rapid backup and recovery (p. 4).

ADVERSARIAL AI

- **1st place, global leaderboard**, AI Jailbreakathon by chat.win: an LLM prompt attack-and-defence contest, won with the fewest tokens ($250 prize).
- **Fool The LLM**: built and ran a prompt-injection contest platform (p. 8).
- **CSAW LLM Attack Challenge (LLMAC) ’25**, NYU: participated.
- **Writing**: “Where LLM guardrails break”, a field report from the contest on what cracks guardrail prompts and how to design systems that do not depend on them.

03

Selected projects

Case studies first, then shorter entries. Ownership uses five fixed words: *solo* (all Kartik's work), *lead* (majority author), *co-lead* (a major share), *maintainer* (took over an existing system) and *contributor* (a minority share). The one team project says *team of 3* instead. Source links appear where the repository is public.

GhostShare

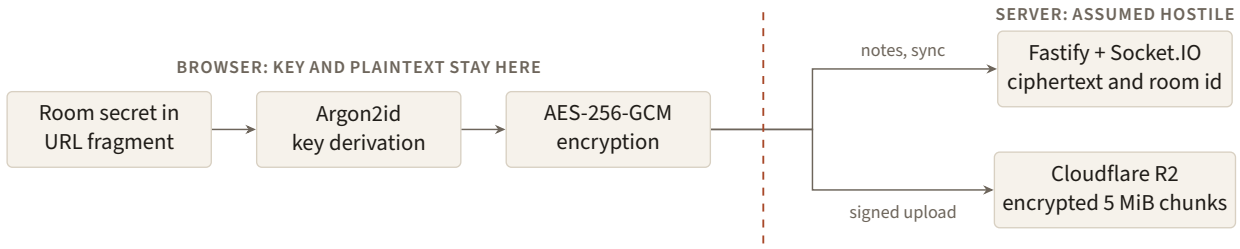
Solo · 2026

End-to-end-encrypted, ephemeral, real-time sharing: anonymous rooms, no accounts, and a server that only ever sees ciphertext.

STACK	TypeScript (strict), Next.js 15, React 19, Fastify 5, Socket.IO 4 with the Redis adapter, PostgreSQL 16 and Drizzle, BullMQ, Redis 7, Cloudflare R2, Web Crypto AES-256-GCM, Argon2id (hash-wasm), pnpm and Turborepo, Docker Compose behind Coolify, Pino, Sentry
LINKS	Live at share.jkartik.in · write-up: "The server can't read this" · project page · source is private

Problem. Most "private" sharing tools still hold plaintext and promise not to read it. GhostShare is designed so that the server cannot.

Threat model. The design assumes the server is hostile: the database leaks, object storage is public, and the operator reads the logs. Each room's secret lives in the URL fragment, which browsers do not send to the server. The browser derives an AES-256 key from that secret with Argon2id, a memory-hard function that makes offline guessing of a weak secret expensive, and seals every note and file with AES-256-GCM, whose authentication tag also exposes tampering by a hostile server.



Encryption happens before anything reaches the network; the dashed line is the trust boundary.

Large files. Files are split into 5 MiB chunks, encrypted in the browser and uploaded straight to Cloudflare R2 with four parallel uploads, exponential-backoff retry and resume. The app server hands out short-lived signed upload URLs and never touches the bytes.

Real time and expiry. Collaboration runs over Socket.IO with a Redis adapter, so it scales horizontally while relaying only ciphertext. Edits made offline queue in IndexedDB and drain on reconnect. Every room has a lifetime chosen at creation (1 day, 3 days, 1 week or 1 month), and an hourly job purges expired rooms and orphaned storage. Helmet and rate limiting guard the API, and backup and restore scripts cover the data that does persist.

Limits, stated in the write-up. Metadata such as timing and blob sizes stays visible to the server, the client code that is served must be trusted, and a room keeps one key for its lifetime; per-room key rotation is the planned next step.

NeuroTechh: clinic appointment platform

Team of 3 · Nov 2026

The first-prize entry at the MAL 4 hackathon: an architecture and prototype in which patients request appointment slots and clinics confirm or decline them.

EVENT	MAL 4 (Mobile Architecture Lab), Paytm, Mumbai, 1 November 2026. First prize, against working professionals and students.
STACK	React Native, Expo, Express, TypeScript, PostgreSQL, Redis, Measure SDK, Docker Compose
LINKS	Source and design documents: github.com/KartikJain14/mal-hackathon · write-up: "We won a hackathon where the code was optional"

What was judged. Architecture was 60% of the score and code was optional. The team of three submitted a one-page problem brief, a full platform design, an HLD, an LLD, 19 Architecture Decision Records and three walkthroughs: the core path, failures and a new requirement. The rule the design protects is that a confirmed slot belongs to exactly one patient and that a cancellation reopens the slot consistently.

Key decisions. One modular API rather than the nine microservices first proposed. Postgres row locks and a partial unique index rather than Redis locks. A transactional outbox rather than Kafka. Pending requests hold no slot, so the race happens when a clinic confirms rather than when a patient asks. Idempotency keys are created on the phone.

Prototype. Built for extra credit: patient, hospital and doctor flows, QR check-in, family profiles and record uploads. Integration tests cover booking and workflows. The booking rule was checked with 2, 50 and 200 concurrent confirms on one slot, and exactly one was confirmed each time.

Limits, stated in the repository. This is a hackathon design and prototype, not a deployed service. The prototype writes outbox rows in the same transaction as each booking change, but the worker that would deliver them exists only in the design. The repository's as-built document records what the code does and where it differs from the design.

Railway Concession

Maintainer · Jul 2026 – present

MPSTME's official online railway-concession system: student submissions, collection-slot booking and tools for the college office.

OWNERSHIP	The original 2025 version was built by the NeuroTechh team. Kartik took it over in July 2026 and now maintains it solo, working with the college administration as its rules change.
SCALE	Used by ~4,000 students
STACK	Next.js, PostgreSQL, Drizzle ORM, Nodemailer
LINKS	Live at concession.mpst.me · write-up: "When a form becomes infrastructure"

Data layer. Migrated the system to PostgreSQL with Drizzle ORM.

Booking rules on the server. A collection date is not accepted just because the browser's date picker allowed it. One shared module validates every date against the lead time, weekends, holidays, the configured end date and a rolling window of open dates, computed in Indian Standard Time whatever the server's own timezone. The client asks the server for the current window only to guide the student; the submission endpoint validates again before writing. The window slides forward on its own, so opening a new day needs no deployment.

Tools for the office. Search by SAP ID with the student's full profile and submission history; mark a form collected for any date; block SAP IDs; manage holidays; tune the booking window; and export to Excel. Students get confirmation emails, and the office gets an automated daily collection-report email. Kartik also patched vulnerable dependencies and improved SEO.

Outcome. A process that ~4,000 students depend on, with rules the office can change from a settings panel rather than waiting on a code change.

Event and payment backends

2024 – 2026

Four backends for NMIMS student events, each with one decision that has to be right: whether money changed hands, or whether a badge was already used.

SYSTEM	EVENT	OWNERSHIP	STACK AND SOURCE
Credits and loyalty	Taqneeq 17.0	Lead author	Flask, Firestore, JWT, Vercel · qrshooter-backend
Delego backend	Mumbai MUN, 2024–25 editions	Lead backend author	FastAPI, PostgreSQL, JWT · mundra
Paid registration	Programming For Everyone	Co-led	Next.js/Node, PostgreSQL, Cashfree · pfe-regi
conc-qr	Social Conclave 2026	Solo	TypeScript, Express, Next.js, PostgreSQL, Sequelize, Docker · conc-qr

Taqneeq 17.0 credits and loyalty backend

Lead author · 2025

Taqneeq, the NMIMS MPSTME tech fest, draws thousands of attendees and runs a fest-wide game in which they earn and spend credits. The backend provides phone-OTP login; a role-based (`SALES` / `ADMIN`) credits ledger that records every transaction; referral codes with abuse handling; an admin portal; and a leaderboard cached with a short TTL, so a crowd refreshing at once reads the cache instead of the database.

Delego backend, Mumbai MUN

Lead backend author · 2024–25 editions

The FastAPI and PostgreSQL service behind Delego, the delegate app for Mumbai MUN: passwordless sign-in with JWT magic links and one-time codes, a QR code per delegate for check-in and meals, organizer CSV import and export, rate limiting for the morning check-in rush, and scheduled database backups. The app shipped on the App Store and Play Store for 600+ delegates; those listings have since been taken down.

Paid event registration

Co-led · 2025

Paid registration for Programming For Everyone. A registration is confirmed only when Cashfree's HMAC-verified webhook says the payment cleared, never on a browser redirect, so a closed tab cannot leave someone charged without a ticket. QR tickets are emailed on confirmation, and gate check-ins are idempotent, so a re-scan cannot admit someone twice. An admin sales dashboard, Google Sheets sync, magic-link login and role-based access complete it.

conc-qr

Solo · used 25 Jan 2026

QR access control for 500+ delegates across Social Conclave's checkpoints, built solo in under two weeks. Whether a scan is new is decided by a PostgreSQL `UNIQUE(participant, access_type, day)` constraint: the endpoint attempts the insert and reads the result, so two simultaneous scans cannot both succeed. Which roles may scan at which checkpoint lives in an access matrix (day → checkpoint → roles) that admins edit mid-event without a redeploy. Staff scan with their phone cameras, with manual entry as a fallback for damaged badges. Write-up: "What conc-qr taught me about myself".

Fool The LLM

Solo · 2026

A prompt-injection contest platform: each level is a Gemini model guarding a secret codeword, and players try to talk it into leaking.

EVENT	Started by Kartik as ACM Webmaster; run by ACM and GDG MPSTME. ~70 participants in the most recent edition.
STACK	TypeScript, Next.js, Express, Prisma, PostgreSQL, Gemini
LINKS	Write-up: "Where LLM guardrails break" · source is private

Design. Levels wrap the model in progressively tighter instructions. The model's own safety filters are deliberately off, because the guardrail prompt is the game, and that is the surface real prompt injection

attacks. Responses stream token by token; an organizer console manages levels and participants, shows chat logs live, and can force-unlock or reset a stuck level. Sign-in uses JWT and magic links, and per-player token rate limits stop one player from spending the event's model budget.

What it taught. The write-up draws the defensive lessons: keep secrets out of the model's context, treat model output as untrusted input, and put the model behind an authorization boundary rather than asking it to be one.

bad-db

Solo · 2026

A database storage engine written from scratch in Java, to understand what happens between INSERT and the disk.

STACK	Java, `RandomAccessFile`, a hand-written B-Tree, its own query language (BQL)
LINKS	Source: github.com/KartikJain14/bad-db · write-up: "A B-Tree on disk"

Records live in a binary file addressed by byte offset. A hand-written B-Tree, with node splitting and `insertNonFull`, maps primary keys to offsets and is rebuilt from the file on open, so the file stays the source of truth. Insert, read, update, delete and upsert are supported; deletes write tombstones, and compaction copies live records into a fresh file to reclaim the space. Older file formats upgrade automatically. On top sits BQL, a small query language with its own tokenizer, parser and interactive CLI. The write-up is explicit that the engine is single-threaded and has no write-ahead log yet.

Self-hosted infrastructure for ACM MPSTME

Solo · 2025 – present

Infrastructure Kartik set up and runs for the ACM student chapter and other college committees, first as Webmaster and now as Chairperson.

- **Platform.** Coolify as a self-hosted deployment platform. It hosts mpstmeacm.com, concession.mpst.me and share.jkartik.in.
- **Mail.** A Mailcow mail server with SPF, DKIM and DMARC configured by Kartik, shared by 7–9 college committees with good deliverability. Every ACM member gets an `@mpstmeacm.com` mailbox on it, at no licence cost.
- **Code.** The chapter's GitHub organisation, GitHub Apps and CI/CD.
- **Community.** An ACM Discord server with custom bots that onboard members and grant channel access by position and department.
- **Members-only Minecraft server (2026–27)**, gated by Discord linking through the ACM-only bot.

MORE PROJECTS

sports-live: tournament auction and scoring platform — Solo · freelance · Aug – Oct 2026

Used at ELEV8; covered with the freelance work on p. 4.

meven: event and facility requisition portal — Solo · in development

A portal for MPSTME that replaces a paper approval chain (faculty mentor → institutional approver → admin allotment → Joint Registrar escalation) with numbered permission documents and a full audit trail. One application covers rooms, refuge areas, canopies, guests and participants, each decided separately. A pnpm monorepo: Next.js, an Express 5 API with a background worker, PostgreSQL, Redis for queues and rate limiting, Cloudflare R2, SMTP, and Docker with Coolify on Ubuntu.

GrouPay: multi-tenant billing for housing societies — Solo · 2025

Three tiers of tenancy (super-admin → channel admin → residents) with the tenancy checks in middleware, so one society cannot read another's ledger. Invoices with public invoice links, an audit log, and OTP plus password sign-in. About 15,000 lines in a pnpm monorepo with 8 migrations; Next.js, Express, Drizzle and PostgreSQL. Source: github.com/KartikJain14/groupay.

ballast: treasury engine for a digital bank

Solo · hackathon build · Oct 2026

A "reserve juggler" built during a hackathon. Each hour it decides how much cash a bank moving ~$20M an hour should keep liquid and how much to sweep into yield tiers, with a cost-ordered fallback funding waterfall. It replays 5 policies and runs stress tests, including an SVB-style bank run; 52 tests check that no decision looks ahead and that the ledger reconciles. Python, with a web app, a Streamlit console and optional Gemini summaries. Source: github.com/KartikJain14/ballast.

flap: an arcade game played over SSH

Solo · 2026

A Flappy-style game that runs in the terminal: `ssh jkartik.in` and play. Written in Go. The game state lives on the server, so clients cannot submit scores; the terminal receives a cell-diffed stream of updates, sprites are half-block pixel art, and a leaderboard ranks players. Source: github.com/KartikJain14/flap.

captr, usb-ai and hindcribe

Solo · 2024 – 2026

captr (2025) turns a topic into a research document: Gemini writes YouTube search queries, yt-dlp and transcripts locate the highlights, FFmpeg pulls key frames at those timestamps, and the result is a Markdown document the user can chat with (FastAPI, MongoDB, JWT and bcrypt, Resend, Next.js). **usb-ai** (2026) is a fully offline coding assistant in one Windows executable: a quantized Gemma-2-2B model embedded in a PyInstaller one-file build and run with llama.cpp, with a streaming Markdown terminal UI and no network code at all (source). **hindcribe** (2024) does offline Hindi speech recognition and Hindi-to-English translation for the DARPG 2024 public-grievances hackathon, where citizen audio could not go to a cloud API: FFmpeg, Vosk, Argos Translate and Flask, with unit tests for each stage (source).

Discord automation bots

Solo · 2024 – 2026

A Taqneeq bot that onboarded 1,100+ hackathon participants with unique invites, real-name renaming and automatic team channels; an ACM onboarding bot with automatic roles and acceptance emails; a custom bot for the MUN Society; and the ACM server bots that grant channel access by position and department.

jkartik.in build and deploy pipeline

Solo · 2026

An Astro static site shipped as a multi-stage Docker image (Node build → nginx-alpine, ~97 MB). GitHub Actions with a buildx cache pushes `:sha` and `:latest` tags to GHCR, and Watchtower pulls new images to deploy, so CI holds no SSH secrets. nginx with certbot TLS, an RFC 9116 `security.txt`, JSON-LD, and Lighthouse scores of 96 / 100 / 100 / 100. Source: github.com/KartikJain14/jkartik.in.

Smaller contributions

An adaptive AI department-classifier quiz for Sattva, Taqneeq and Social Conclave, whose questions adapt to earlier answers (solo or lead). Contributed to a certificate portal: Docker and Coolify deployment, email sending and fixes. Set up and deployed mpstme.pics: repository scaffold, Docker, CI/CD with GitHub Actions and Watchtower, and several access-control endpoints. template-portfolio (lead), a fill-one-file portfolio template used by PFE participants.

04

Leadership and community

All of these organisations are at SVKM's NMIMS MPSTME, Mumbai. In 2025–26 Kartik headed the technical department of four of them at once (Taqneeq, the MUN Society, Social Conclave and Sattva) while serving as ACM's Webmaster.

ORGANISATION	ROLE	DATES
ACM MPSTME	Chairperson	Jun 2026 – present
	Supercore: Webmaster	Jul 2025 – Jun 2026
	Sub-Head Technicals (Competitive Programming)	Jul 2024 – Jul 2025
Taqneeq (tech fest)	Head of Technical Department, Taqneeq 18.0	Aug 2025 – May 2026
	Technical Sub Head, Taqneeq 17.0	Oct 2024 – May 2025
MUN Society MPSTME	Head of Department: Technical	Jun 2025 – May 2026
	Executive (Technicals)	Jul 2024 – Jun 2025
Social Conclave	Head of Technical Department	Aug 2025 – May 2026
Sattva (cultural fest)	Head of Technical Department	Sep 2025 – May 2026
NeuroTechh	Maintainer, Railway Concession system	Jul 2026 – present

ACM MPSTME

Jul 2024 – present

Chairperson. Chairs a chapter of ~250 members, 50 of them core members. Led recruitment and restructured the chapter's departments, creating new ones. Grew Programming For Everyone (PFE) to a record ~500 learners and ran it for the first time as a 5-day program open to students from other colleges, with workshops on C, Python, cybersecurity, web development, web hosting, DSA, AI and Git/GitHub; Kartik personally mentored the C, cybersecurity and web-hosting sessions. Partnered with SNDT ACM-W. Semicolon, the chapter's introductory event, drew record footfall over two days, and an alumni networking event brought together 70–80 members and alumni.

Webmaster. In a committee of 120+, built the chapter's self-hosted infrastructure (p. 9): Coolify, and a Mailcow mail server with SPF, DKIM and DMARC that is now shared by 7–9 college committees. Set up the GitHub organisation, GitHub Apps and CI/CD. Wrote problem statements for an overnight hackathon and judged and mentored at it, set competitive-programming problems, and started Fool The LLM (p. 8).

Sub-Head Technicals. Set up the chapter's Discord server and its onboarding bot (automatic roles, real-name renaming, acceptance emails). Taught first-years C++ at PFE-24 by building a command-line game. Contributed to a certificate portal with verification and add-to-LinkedIn, set competitive-programming contest problems, and designed a CTF for freshers.

Taqneeq

Oct 2024 – May 2026

Head of Technical Department, Taqneeq 18.0. Led the revamp of the fest app and backend, including leading the team that built photo discovery with facial recognition (attendees upload event photos and find the ones they appear in), and gamified points redeemable for coupons and merchandise. The department's work also covered the Taqneeq website (taqneeqfest.com), contributions to the Students' Council website, the department classifier and the fest's Discord ecosystem, with the fest's email and servers running on the ACM self-hosted infrastructure. Kartik set problems for and mentored at CyberCypher, Taqneeq's cybersecurity hackathon.

Technical Sub Head, Taqneeq 17.0. Lead author of the credits and loyalty backend (p. 8); integrated it with the Flutter app; built the Discord bot that onboarded 1,100+ hackathon participants; and wrote a certificate-generation script.

MUN Society MPSTME

Jul 2024 – May 2026

The society that runs Mumbai MUN (munsocietympstme.com).

Head of Department: Technical. Revamped the society's app, Delego, and its backend; migrated the database from SQLite to PostgreSQL; set up transactional email; maintained the servers; and built the society's custom Discord bot and Muntree, a Linktree-style site for the MUN community.

Executive (Technicals). Built magic-link password reset, QR scanning and a food-usage page; documented the backend; fixed Flutter build and SDK issues; and ran a GitHub and LinkedIn workshop.

Social Conclave, Sattva and NeuroTechh

2025 – present

Social Conclave, Head of Technical Department. Built conc-qr solo (p. 8): QR access control for 500+ delegates, used at the event on 25 January 2026.

Sattva, Head of Technical Department. The fest website, registration forms and CI workflows; an AI-enabled adaptive department classifier whose questions adapt to earlier answers; and promo reels.

NeuroTechh. Maintains the Railway Concession system used by ~4,000 students (p. 7).

TEACHING AND MENTORING

- Mentor at Programming For Everyone for C, cybersecurity and web hosting; C++ teacher at PFE-24.
- Hackathon judge and mentor at ACM's overnight hackathon; problem setter and mentor at CyberCypher.
- Problem setter for competitive-programming contests; designed a CTF for freshers.
- Ran a GitHub and LinkedIn workshop for the MUN Society.
- Built template-portfolio, a fill-one-file portfolio template used by PFE participants.

05

Achievements

First prize, team of 3	MAL 4 hackathon (Mobile Architecture Lab, Paytm, Mumbai, 1 November 2026), against working professionals and students; supported by Measure, Google for Developers, Paytm and Ocean AI; ₹20,000 prize (p. 7)
1st place, global leaderboard	AI Jailbreakathon by chat.win: LLM prompt attack and defence, won with the fewest tokens; $250 prize
$1,500 bug bounty	Awarded by Meta for a privilege-escalation vulnerability in WhatsApp (2022)
1st place, team win	CodeForge 2026 (Taqneeq × FOSS MPSTME): built and pitched a NIFTY options-analytics platform with a 3-person team
Rank 1	Smart India Hackathon, internal college round
Finalist, top 15 of 230 teams	GDG AI Hackathon
Participant	CSAW LLM Attack Challenge (LLMAC) '25, NYU
Participant	DARPG 2024 hackathon, with hindcribe (p. 10)

06

Skills, with evidence

Every skill below points to where it was used. **Strong** means used in production; **working** means used in a project; **light** means occasional use.

LANGUAGES		
TypeScript, JavaScript	Strong	VisionRise LMS, GhostShare, meven, GrouPay, conc-qr
Python	Strong	Delego (FastAPI), Taqneeq (Flask), ballast, captr, hindcribe, usb-ai, bots
SQL	Strong	PostgreSQL across projects
Java, Go	Working	bad-db (Java), flap (Go)
C, C++	Working	Taught C++ at PFE-24; mentored C at PFE
Bash	Working	Backup and restore scripts, server operations
Dart / Flutter	Light	Delego and Taqneeq app integration and fixes
BACKEND		
Node.js, Express	Strong	LMS, meven, conc-qr, GrouPay, Fool The LLM
FastAPI, Flask	Strong	Delego backend (FastAPI); Taqneeq backend, hindcribe (Flask)
Next.js	Strong	GhostShare, Railway Concession, conc-qr, sports-live, meven, PFE registration
REST API design	Strong	276-route LMS backend
Fastify, WebSockets / Socket.IO	Working	GhostShare
Auth: JWT, OAuth 2.0 / Google Sign-In, OTP, magic links	Strong	LMS, Delego, Fool The LLM, PFE registration
RBAC and access control	Strong	LMS, conc-qr, GrouPay, Taqneeq
Payments: Cashfree HMAC webhooks, UPI verification	Working	PFE registration, LMS
Queues and background workers: BullMQ	Working	GhostShare, meven
ORM: Drizzle	Strong	LMS, GhostShare, GrouPay
ORMs: Prisma, Sequelize	Working	Fool The LLM (Prisma), conc-qr (Sequelize)
Validation: Zod	Strong	LMS, GhostShare
Testing: unit and integration tests, Vitest	Working	LMS, ballast (52 tests), sports-live
System design: HLD, LLD, Architecture Decision Records	Working	NeuroTechh (p. 7)
DATA AND CLOUD		
PostgreSQL	Strong	Most projects; SQLite to PostgreSQL migration for the MUN Society
Redis	Working	GhostShare (pub/sub, presence, queues), meven
MongoDB	Working	captr, freelance work
Firebase / Firestore, FCM	Working	Taqneeq backend; LMS push notifications
Supabase, Neo4j	Working	sports-live (Supabase); Semgrep visualisation at R.U.D.R.A (Neo4j)
Google Cloud Storage: signed URLs, backups	Strong	LMS media; VisionRise server backups
Cloudflare R2, Vercel	Working	GhostShare and meven (R2); Taqneeq backend (Vercel)

DEVOPS AND OPERATIONS		
Docker, Docker Compose	Strong	Most projects
Linux server administration (Ubuntu 24.04 LTS), hardening	Strong	VisionRise server, ACM infrastructure, freelance
Coolify (self-hosted PaaS)	Strong	VisionRise, ACM, GhostShare, meven
CI/CD: GitHub Actions, GHCR, Watchtower	Strong	jkartik.in, mpstme.pics, ACM GitHub organisation
Nginx, Caddy, reverse proxies, TLS (certbot)	Working	jkartik.in, LMS deployment configs
Email infrastructure: Mailcow, SMTP, SPF/DKIM/DMARC	Working	ACM mail server shared by 7–9 committees
Backups and disaster recovery	Strong	VisionRise GCS backups, freelance ransomware recovery, Delego scheduled backups
Logging and observability: Pino, Sentry	Working	GhostShare, VisionRise
Git, GitHub organisation administration	Strong	ACM, VisionRise
Google Workspace administration	Working	VisionRise (35+ accounts)
MOBILE		
React Native, Expo (EAS, OTA updates), Google Play publishing	Strong	VisionRise: two apps from one codebase
SECURITY		
Web application penetration testing, VAPT	Strong	R.U.D.R.A, disclosures (p. 5)
OWASP Top 10: XSS, broken access control, LFI, information disclosure	Strong	Disclosures and the banking-client findings
Manual code review	Strong	R.U.D.R.A banking-client findings
Semgrep (custom module), Nuclei (custom templates), BBOT	Strong	R.U.D.R.A
Burp Suite	Strong	Web-app testing and bug bounty work
Nmap	Working	Reconnaissance and network scanning
Red and purple teaming: Sliver C2	Working	R.U.D.R.A
Responsible and coordinated disclosure	Strong	Meta, MSRC, CERT-In
Incident response	Working	Freelance ransomware recovery
Applied cryptography: AES-GCM, Argon2id, AES-128 HLS, HMAC, signed URLs, SHA-256	Strong	GhostShare, LMS, PFE registration
CTF design	Working	ACM freshers' CTF
AI AND LLMS		
LLM red teaming, prompt injection	Strong	chat.win win, Fool The LLM, CSAW LLMAC '25
Gemini API	Strong	LMS auto-quiz, Fool The LLM, captr, ballast
llama.cpp, local LLMs (GGUF); Vosk, Argos Translate	Working	usb-ai; hindcribe

07

Education

B.Tech, Information Technology

Jul 2024 – Apr 2028 (expected)

SVKM's NMIMS Mukesh Patel School of Technology Management & Engineering (MPSTME), Mumbai

CGPA	8.34 / 10.0 (as of October 2026)
LANGUAGES	English (full professional); Hindi and Marwari (native)

08

Links

Every web link below was checked on 2 November 2026. Source code for client and employer work, and for GhostShare, Fool The LLM, meven, sports-live and captr, is private.

KARTIK

Website	jkartik.in
Research	jkartik.in/research
Projects	jkartik.in/projects
GitHub	github.com/KartikJain14
LinkedIn	linkedin.com/in/kartikjain1410
Email	career@jkartik.in

LIVE SYSTEMS

GhostShare	share.jkartik.in
Railway Concession	concession.mpst.me
Rise Academy	Google Play listing
eGuru	Google Play listing
VisionRise sites	risetechnosoft.com riseinstitute.tech · eguru.app
flap	`ssh jkartik.in`

ORGANISATIONS

ACM MPSTME	mpstmeacm.com
NeuroTechh	neurotechh.mpst.me
Taqneeq	taqneeqfest.com
MUN Society	munsocietympstme.com

SOURCE CODE (GITHUB.COM/)

KartikJain14/bad-db
KartikJain14/mal-hackathon
KartikJain14/qrshooter-backend
munsoc-mpstme/mundra
MPSTME-ACM/pfe-regi
Social-Conclave/conc-qr
KartikJain14/groupay
KartikJain14/ballast
KartikJain14/flap
KartikJain14/usb-ai
KartikJain14/hindcribe
KartikJain14/template-portfolio

WRITING (JKARTIK.IN/BLOG)

The server can't read this
When a form becomes infrastructure
What conc-qr taught me about myself
Where LLM guardrails break
A B-Tree on disk