Rajendra Bist

Full-Stack Web Developer & Backend Engineer

+977 9815625989 | rajendrabist396@gmail.com | linkedin.com/in/rajendra-bist | github.com/rajendrabist07 | rajendra-bist.vercel.app

Summary

Full-stack engineer with a backend-first mindset, trained at Vcare Technical Institute and currently pursuing B.Tech in Computer Science at Indira Gandhi National Open University (IGNOU). Architects and ships AI-integrated, production-grade systems — from autonomous LLM tool-calling agents and GitHub App integrations to streaming LLM pipelines and multi-model AI orchestration — with deliberate attention to API design, database schema modelling, security hardening, and observability. Four deployed products in active production. Seeking backend-heavy full-stack or AI engineering roles.

Technical Skills

Languages: JavaScript (ES6+), TypeScript, Python (basic), HTML5, CSS3
Frontend: React 19, Next.js 15 (App Router), Tailwind CSS v4, Framer Motion, Recharts, Responsive Design
Backend: Node.js, Express.js, REST APIs, Server-Sent Events (SSE), WebSockets, JWT Auth, Zod Validation
AI / LLM: Groq API (Llama 3.3 70B), Gemini API, Multi-tier LLM Fallback, Prompt Engineering, Streaming Interfaces
Databases: Supabase (PostgreSQL), MongoDB Atlas, Mongoose, pgvector, Upstash Redis
Tools & DevOps: Git, GitHub Actions (CI/CD), Vercel, Clerk Auth, Octokit, Vitest, Docker (learning), Linux CLI

Projects

DevGuard AI — *Autonomous PR Security & Code Review Agent* | Live | GitHub **2025**

- **Autonomous Tool-Calling Agent:** Engineered an LLM orchestration loop where the model actively invokes three diagnostic tools — an AST-based static security linter (SQL injection, XSS, unhandled promises), an OSV.dev CVE dependency scanner, and a programmatic Vitest/Jest test runner — before synthesising severity-tagged findings with one-click copyable inline PR patches, eliminating hallucination-based review feedback with empirical, tool-verified evidence.
- **Multi-Tier LLM Fallback Pipeline:** Designed a resilient AI routing layer: Groq Llama 3.3 70B as primary, automatic failover to Gemini 2.5 Flash on HTTP 429 rate-limit, and a deterministic rule engine as final safety net — ensuring 100% review delivery under any model availability condition, with explicit model attribution in the UI transparency panel.
- **GitHub App Integration & Webhook Security:** Built a verified GitHub App using Octokit REST and App Auth that receives PR webhook events, validates HMAC-SHA256 signatures, and posts inline review comments and suggested code patches via the GitHub Checks API.
- **Security Hardening:** Upstash Redis sliding-window rate limiting (5 req/10 min/IP), Zod schema validation on all public endpoints with 100KB payload cap, HTTP security headers (X-Frame-Options: DENY, CSP, nosniff), zero-eval AST sandbox preventing RCE, and a regex-based secret scrubber redacting keys (sk_live_*, ghp_*, AIza*) from JSON logs before Vercel Log Stream output.
- **Production Observability & CI:** 27-passing Vitest test suite covering tool invariants, SVG badge geometry, PDF/Markdown export, and LLM attribution; GitHub Actions pipeline enforcing typecheck, lint, test, and build on every PR; Sentry exception tracking; live health endpoint at `/api/health`; Dependabot weekly dependency scanning.

EduMethod AI — *AI-Powered Personalised Learning Platform* | Live | GitHub **2025**

- Architected a full-stack AI learning platform converting book photos or syllabus text into structured 7-day study plans with spaced repetition scheduling, adaptive quizzes, and a conversational doubt-coach; dual-LLM pipeline: Groq (Llama) for low-latency plan generation, Gemini API for multimodal image extraction.
- Production data layer: Clerk authentication with protected routes; Supabase (PostgreSQL + pgvector) for user progress tracking and vector-based content retrieval; Upstash Redis for API response caching and rate limiting.

SocraticAI — *Socratic Method AI Learning Assistant* | Live | GitHub **2025**

- Chat-first AI assistant enforcing Socratic dialogue via custom system prompt engineering; Gemini API streamed token-by-token via Web API ReadableStream in a Next.js 15 serverless route; MongoDB Atlas session logging with TTL auto-expiry and in-memory rate limiting protecting free-tier quota.

Developer Portfolio — *Streaming AI Chat Agent* | Live **2025**

- Floating Gemini-powered AI agent streaming context-aware portfolio responses via ReadableStream; dark/light theme, Framer Motion scroll-reveal animations, contact form with MongoDB persistence, Lighthouse Performance 90+.

Education

B.Tech in Computer Science & Engineering (Ongoing)
Indira Gandhi National Open University (IGNOU)

Full-Stack Web Development
Vcare Technical Institute *Completed 2025 — Next.js, Node.js, React, MongoDB, Express.js, REST APIs, System Design*

+2 Science
KMC — Kailali Multiple Campus

Additional

Currently: Extending DevGuard AI with real GitHub App install flow and authenticated analytics dashboard
Focus Areas: LLM agent orchestration, system design, microservice architecture, PostgreSQL query optimisation
Location: Nepal (UTC+5:45) — remote worldwide; open to onsite/hybrid in Nepal or India
Languages: Nepali (Native), Hindi (Fluent), English (Professional)