

Saroath Farvees



+94 75 672 0854 | jmsaroathfarvees@gmail.com
farvees.xyz | linkedin.com/in/saroath-farvees | github.com/mrfarvees-official

Professional Summary

Full-stack developer specializing in multi-tenant architecture and access control (RBAC/PBAC), with production experience across backend APIs, cloud infrastructure, and DevOps.

Technical Skills

Languages: Java, Python, PHP, TypeScript, JavaScript, SQL, HTML/CSS, Bash
Frontend: React, Next.js (App Router), Tailwind CSS, Server-Sent Events, Responsive UI
Backend & Databases: Spring Boot, Laravel, Node.js, REST APIs, MySQL, SQLite, Redis, Prisma ORM, Drizzle ORM, Zod, FFmpeg
AI & Browser Automation: RAG (embeddings, hybrid retrieval), LLM Tool Calling, Streaming APIs (NDJSON), Chrome Extensions (Manifest V3), Prompt-Injection Defenses
Cloud & DevOps: AWS S3, AWS SES, Oracle Cloud Infrastructure (OCI), Docker, Docker Compose, Git, CI/CD, Nginx, PM2, Certbot, Ubuntu Server
Architecture & Security: Multi-Tenancy, RBAC, PBAC, Audit Logging, Database Isolation, Backup & Restore, SSRF Protection, Rate Limiting, Session Authentication, SSL/TLS, Firewall & Network Configuration
Testing & Tooling: Vitest, ESLint, TypeScript (strict), dependency-cruiser, esbuild, pino, Prometheus
Integrations: Resend, Cloudinary, SMS Services, NVIDIA-hosted LLM APIs (OpenAI SDK), Gmail/Calendar/Meet (via browser extension)

Experience

NexFort (Pvt) Ltd.

Software Engineering Intern

July 2025 – Present

Galle, Southern Province

- Developed a multi-tenant real estate platform with role-based access control (RBAC), REST APIs, MySQL, and Redis, enforcing tenant-scoped data access and supporting secure, scalable operations for multiple organizations on a single system.
- Implemented tenant-aware features from database schema to REST endpoints, applying the platform's RBAC rules so each organization could reach only its own data and the actions its roles permit.
- Integrated AWS S3, AWS SES, and FFmpeg to deliver cloud file storage, transactional email, and media processing, covering the platform's document, notification, and media-handling requirements.
- Managed development workflows using Git issues for task tracking and contributed to CI/CD pipelines, improving deployment consistency and team collaboration across releases.

Freelance

Software Engineer – Debt Recovery Management System

May 2026 – Sep 2026

Remote – Client: Uganda

- Architected a multi-tenant debt-recovery ERP backend (**Laravel, PHP, MySQL**) with a **database-per-organization** design, custom tenant provisioning and a migration runner, serving 300+ REST endpoints with isolated data for every client.
- Secured the platform with **JWT auth and policy-based access control (PBAC)** that denies unmapped routes by default, plus a **maker-checker approval workflow**, a correlated audit trail and an append-only, secret-redacting activity log for full accountability.
- Built the recovery workflow end to end: installment schedules, payments, agent pledges, SMS/email reminders and bulk campaigns, configurable **credit-risk reports**, Excel imports (20+ entities), and a resilient **queued backup/restore** system with per-tenant and system-wide recovery, storage quotas and an admin-portal dashboard.
- Containerized with **Docker** and deployed on **Oracle Cloud** (Ubuntu, Nginx, Certbot/TLS, firewall and network routing) with **GitHub Actions CI/CD**; integrated **Resend** for transactional email and used Redis for caching.

Projects

PeopleOps — Multi-Tenant HR Platform | *Next.js, TypeScript, Prisma, MySQL*

[Site](#) | [GitHub](#)

- Built a multi-tenant HR platform covering employees, leave, attendance, recruitment, and payroll on a layered architecture, with dependency-cruiser rules enforcing domain, application, and infrastructure boundaries across 165 modules.
- Implemented a default-deny PBAC engine with attribute conditions and global deny policies (self-approval, closed payroll runs, cross-tenant access); every action runs a permission check, then the operation, then an audit entry.
- Designed a configuration-driven HR module framework that provides CRUD, workflows, soft delete, SQL-level row restriction, and sensitive-field masking from a single config per module, avoiding per-module security code.
- Added table-driven import/export (CSV and JSON) and checksummed backup/restore with transactional dry-run previews that roll back on any row error, a safety backup taken before every restore, and tenant checks that stop an import from overwriting another company's data; verified with 159 Vitest tests across 13 files, including a role-by-resource policy matrix, and deployed behind an HTTPS Nginx proxy.

ReconLite — Passive Website Security Inspector | *Next.js, TypeScript, Node.js, SSE*

[Site](#) | [GitHub](#)

- Built a passive security scanner running eight parallel checks (HTTP headers, cookies, CORS, TLS, DNS email policy, technology fingerprinting, CVE matching, database error leaks) and streaming findings to a terminal-style UI over Server-Sent Events.
- Engineered SSRF and DNS-rebinding protection with a custom DNS lookup that rejects private, loopback, and link-local addresses at connect time, plus HTTPS-only requests, capped response bodies, and per-module timeouts.
- Added abuse controls including server-enforced consent, per-IP rate limiting (5 scans per 10 minutes, one concurrent scan), and cached, rate-aware NVD lookups with results worded as “verify patch level” to avoid false-positive claims.
- Shipped to production with PM2, Nginx, and Let's Encrypt using a re-runnable deploy script; deliberately excluded active exploitation testing to keep the tool safe to expose publicly.

Chatbot — RAG Assistant | *Next.js, TypeScript, SQLite, Drizzle ORM*

[Site](#) | [GitHub](#)

- Built a retrieval-augmented chatbot with heading-aware chunking, 2048-dimension embeddings, and hybrid cosine plus keyword ranking over SQLite, answering only from ingested documents and streaming responses as NDJSON.
- Created an admin knowledge portal with hash-based, idempotent ingestion that embeds before replacing, so a failed update never destroys the previous version of a document.
- Implemented local authentication (scrypt password hashing, hashed httpOnly session tokens), per-IP rate limiting, and strict request validation protecting a paid model API from unauthenticated use.
- Developed a tool-routing layer that sanitizes and narrows around 40 browser tools per request and forces the first tool when intent is clear, keeping prompts small and behavior unit-testable with Vitest.

Site Agent — Chrome Extension for AI Browser Actions | *TypeScript, Manifest V3, esbuild*

[Site](#) | [GitHub](#)

- Built a Manifest V3 extension exposing 42 Gmail, Calendar, and Meet tools to a chatbot through an origin-checked postMessage bridge, using the visitor's own signed-in session so no OAuth token or mailbox access ever reaches the server.
- Designed risk-based execution (read, write, irreversible) with an extension-owned, default-deny approval popup (60-second timeout), a public tool allowlist, a global pause switch, per-tool toggles, and an audit log that stores argument names only.
- Implemented a site-adaptor architecture with health checks and a Gmail Send matcher that refuses to act when zero or multiple Send controls match, preventing a message from going out on the wrong draft.
- Handled service-worker restarts by persisting state in session storage and adding timeouts to all tab messaging; covered the tool contract, discovery, and send-control matching with Vitest.

Education

University of Kelaniya

Expected Nov 2026

Bachelor of Information and Communication Technology (Honours) (GPA: 3.50 / 4.00)

Kelaniya, Western Province

- **Relevant Coursework:** Data Structures and Algorithms, Database Systems, Structured Programming, Computer Architecture and Organization, Operating Systems, Data Communication and Networks