

Rakshit Rameshbabu

Software & Security Engineer

Chennai, Tamil Nadu, India

[in rakshit-rameshbabu](#) | [rakshit-737](#) | [rakshitoff@gmail.com](#) | [+91 87787 80084](#)

SUMMARY

B.Tech (Cyber Security) student at VIT Chennai with strong CS fundamentals and hands-on experience building full-stack products and backend systems — REST APIs, ML-driven detection pipelines, and security tooling — owned end-to-end from requirements to CI-tested deployment. AI-native developer who uses LLM coding tools daily to prototype, debug, and ship faster. Seeking an SDE internship building scalable, data-intensive systems.

TECHNICAL SKILLS

Languages	Python, Java, C/C++, TypeScript, JavaScript, SQL, HTML/CSS
Core CS	Data Structures & Algorithms, OOP, Operating Systems, Computer Networks, DBMS
Backend & Frameworks	FastAPI, Node.js/Express, REST APIs, SQLAlchemy, React, React Native (Expo), Next.js, scikit-learn
Databases & Infra	PostgreSQL, Redis, SQLite, Docker, Git/GitHub, GitHub Actions (CI/CD), AWS (Basics), Linux
AI-Native Workflow	LLM-assisted development (Claude Code) for prototyping, debugging, test generation, and code review
Security	Wireshark, Nmap, Burp Suite, Metasploit, Nessus, ClamAV; network/web security, threat detection, supply-chain security

EDUCATION

2024 – Present	B.Tech in Computer Science Engineering (Cyber Security), VIT Chennai	CGPA: 9.07 / 10
2023 – 2024	Class XII (PCM-CS), National Public School	93.6%
2021 – 2022	Class X, National Public School	96.2%

PROJECTS

Warden – Software Supply-Chain Firewall	GitHub	Jul 2026
--	------------------------	----------

Behavioral firewall for open-source dependencies; fuses rule-based and ML signals into a 0–100 risk verdict.

- Built a supply-chain firewall that statically analyzes PyPI packages (no code execution), fusing six independent analyzers (metadata/provenance, AST behavior, install-time execution, typosquatting, obfuscation, IOC matching) into explainable, per-signal risk evidence.
- Trained and served a calibrated ML model (RandomForest + IsolationForest) fused with a tiered rule engine into a 0–100 score, enforcing allow/warn/block policy via a REST API, a CLI/CI gate, and a React dashboard.
- Hardened against attacker-authored input (anti-zip-bomb and anti-path-traversal extraction) with JWT auth + refresh rotation, argon2id hashing, RBAC, rate limiting, and an append-only audit trail; 40 automated tests in CI.

Tech: Python 3.12, FastAPI, SQLAlchemy 2, scikit-learn, PostgreSQL, Redis, React 18, TypeScript, Docker

PlantPal+ – Full-Stack Daily Wellness & Plant-Care Platform	GitHub	2026 – Present
--	------------------------	----------------

Cross-platform app unifying plant care, fitness, and nutrition tracking in a single daily dashboard with streaks, reminders, and cloud sync.

- Architected a TypeScript monorepo — React Native (Expo) mobile app, React + Vite web app, and Node.js/Express REST API on PostgreSQL — with a node-cron reminder engine, an 11-table JWT auth service with a custom migration runner, and GitHub Actions CI.
- Owned the project end-to-end through a full SDLC: 36 requirements documents (228 functional requirements, 111 NFRs, 119 user stories, 89 use cases), system architecture, database schema, and API specification before implementation.
- Built and unit-tested a shared scientific domain layer (Mifflin-St Jeor/TDEE energy models, workout energy/1RM/volume, watering intervals) and offline-light sync via an append-only event log with client-generated UUID idempotency keys.

Tech: TypeScript, React Native (Expo), React, Node.js/Express, PostgreSQL, node-cron, GitHub Actions

ONGOING RESEARCH

Proactive Feasibility Scheduler	GitHub	2026 – Present
--	------------------------	----------------

Evaluation study of ML-based GPU-cluster job scheduling (XGBoost wait-time model + simulation), yielding a proven negative result.

- Built an end-to-end pipeline — discrete-event cluster simulator, an XGBoost wait-time regressor, and a 14-policy benchmark (FCFS, SJF, EASY/conservative backfill, SRPT, HRRN, ML) with Holm-adjusted significance testing.
- Proved the learned scheduler is structurally degenerate: across 45,432 real dispatch instants (zero counterexamples) its queue ordering collapses to a sort by requested job size, established with paired TOST equivalence tests ($p = 2.6 \times 10^{-16}$) rather than difference tests.
- Validated on real supercomputer traces (LANL CM-5, SDSC SP2) via second-exact event replay, showing the simulated ML gain does not replicate; fully reproducible (seeded pipeline, Docker, CI) and written up as a LaTeX manuscript.

Tech: Python, XGBoost, scikit-learn, NumPy/pandas, discrete-event simulation, statistical testing (TOST/Holm), Docker

ACHIEVEMENTS

- First Prize** – Cyber Secure 360 Expo 2025, organized by SCOPE, VIT Chennai (Jun 2025).
- Top 100 Teams** – FarAway Zuup Hackathon (Jun 2026), among ~11,000 participants.