

# Nithin Kumar Reddy

📍 Bengaluru, Karnataka | 📞 +91 7259668646 | ✉️ [nithinkr078@gmail.com](mailto:nithinkr078@gmail.com)  
🌐 [github.com/Nithin078](https://github.com/Nithin078) | 🔗 [linkedin.com/in/r-nithin-kumar-reddy](https://linkedin.com/in/r-nithin-kumar-reddy) | 🌐 [nithin078.github.io](https://nithin078.github.io)

## SUMMARY

- B.Tech CSE student (graduating May 2027) focused on **backend systems**, cloud data platforms, and AI tooling — multi-agent orchestration, autonomous developer tools, and cloud security open source.
- Comfortable owning work end-to-end: design, implementation, testing/CI, debugging, and shipping. Core stack: Python, FastAPI, Azure/Databricks, AWS, GitHub Actions.
- Contributed merged PRs to projects like Prowler and OpenSRE; built larger systems such as Aegis and an agentic ETL platform. Looking for software engineering internships where I can improve reliability, automation, and developer experience.

## EDUCATION

### PES University

Aug 2023 – May 2027

B.Tech, Computer Science and Engineering — CGPA: 7.04/10.0

Bengaluru, Karnataka

**Coursework:** Data Structures & Algorithms, Object-Oriented Analysis & Design (OOAD), DBMS, Operating Systems, Computer Networks, Compiler Design, Cloud Computing, Machine Learning, Big Data, Data Analysis

## TECHNICAL SKILLS

**Languages & Scripting:** Python, C/C++, Java, JavaScript/TypeScript, SQL, Bash  
**Backend & APIs:** FastAPI, Node.js, Express, React, REST, WebSockets, JWT/OAuth  
**Cloud & Data:** Azure (ADF, Functions, ADLS Gen2), Databricks, Apache Spark, Kafka, AWS, Docker, Kubernetes, Linux, Git  
**Databases:** PostgreSQL, MySQL, MongoDB, SQLite, Neo4j, Firebase, Supabase  
**Testing & Tools:** pytest, Selenium, Playwright, unit/integration testing, test design, RCA/debugging, GitHub Actions CI  
**AI/ML:** Multi-agent systems, RAG, LangChain, Hugging Face Transformers, scikit-learn, TensorFlow, OpenCV, CLIP, Pandas/NumPy  
**Enterprise:** SAP S/4HANA Sales & Distribution (SD)

## EXPERIENCE

### Center for Information Security, Forensics and Cyber Resilience (IFSCR)

June 2025 – August 2025

Summer Research Intern — Digital Image Forensics

PES University

- Built **image-source-detection**, a forensics pipeline that finds the original source of images moving through a social-style network using **CLIP** and **pHash** fingerprints plus a **NetworkX** user–post graph (not metadata alone).
- Ranks candidates with combined similarity scores, picks the earliest strong origin by timestamp/path/score, and shows side-by-side evidence in a Tkinter demo for multi-hop cases.
- Added unit tests for scoring and origin logic; used interactive post/follow/trace demos to walk stakeholders through the workflow.
- [github.com/Nithin078/image-source-detection](https://github.com/Nithin078/image-source-detection)

## PROJECTS

### Prowler — Open Source ★ 14.6k | Python, AWS, compliance checks

June 2026 – August 2026

- Contributed to **Prowler**, a widely used open-source cloud security tool that runs automated compliance checks on AWS.
- **PR #12076** (merged): added a check for EC2 instances left stopped longer than an admin-set threshold, helping catch idle cost and compliance issues.
- **PR #12118** (merged): added a SageMaker endpoint-config KMS encryption check with service model support and unit tests.

### Memgraph — Open Source ★ 4.3k | C++, Cypher query planner

July 2026

- Contributed to **Memgraph**, a high-performance graph database, on the Cypher query planner path.
- **PR #4490**: found and fixed a planner SIGSEGV for RETURN-only EXISTS subqueries (null input\_op), so that path no longer crashes.

### OpenSRE — Open Source ★ 10.2k | Python, agent harness, testing

July 2026

- Contributed to **OpenSRE** (Tracer Cloud) with unit tests around the agent conversation harness.
- **PR #4401** (merged): tests for record\_conversation\_turn covering append, window bounds, summary compaction, and in-place updates to reduce regressions in agent state.

- Contributed to **RTK** release automation and CI/CD on a large open-source project.
- **PR #3251**: limited CD/release jobs to the main repo so forks do not trigger noisy or failing release pipelines.

### Aegis Engineer | Python, multi-agent, AST, NetworkX, pytest, CLI/TUI

June 2026 – August 2026

- Building an autonomous software-engineering tool that understands a repo with Python AST, call graphs, and hybrid search, then uses multi-agent workflows to plan and apply changes.
- Solve loop: plan → code → ruff/pytest → security/perf/regression review → optional GitHub PR; retries failed lint/tests and isolates edits with git worktrees/snapshots.
- Ships as a local CLI, TUI, and HTTP API, with a pre-push quality gate (secrets, unit/integration tests, optional lint) and run traces for tool use, latency, and cost.
- [github.com/Nithin078/Aegis-Engineer](https://github.com/Nithin078/Aegis-Engineer)

### Agentic Big Data Pipeline Orchestration | Python, FastAPI, Azure ADF, Databricks, React

June 2025 – May 2026

- Architected a **7-agent** ETL orchestrator (Planner, Resource, Performance Prediction, Monitor, Assurance, Executor, Central Manager) on **Azure Data Factory + Databricks**.
- Trained models for automated compute sizing (workers, DIU, shuffle partitions), cutting over-provisioning by **~30%** in evaluation runs.
- Added real-time monitoring (WebSockets), anomaly detection, and mid-run reallocation, reducing simulated pipeline failure rate by **~25%**; dual-validation Assurance Agent gates plans before execution.
- Built a FastAPI + React dashboard for cost/performance visibility, with structured agent handoffs for failure isolation and re-planning.
- [github.com/Nithin078/Agentic-AI-Driven-Dynamic-Orchestration-of-Big-Data-Pipelines](https://github.com/Nithin078/Agentic-AI-Driven-Dynamic-Orchestration-of-Big-Data-Pipelines)

### DemoGen | Python, Playwright, FastAPI, React, TTS

March 2026 – August 2026

- Built a platform that turns product walkthroughs into narrated demo videos using **Playwright** browser automation, screen capture, TTS, and video assembly.
- Fixed flaky multi-step UI failures (timing, selectors) and headless/bot-detection issues (viewport, user-agent, wait strategies) so automated runs complete more reliably.
- FastAPI + React control plane for job status, clear failure points, and downloadable outputs so a failed step is easy to find and re-run.
- [github.com/Nithin078/DemoGen](https://github.com/Nithin078/DemoGen)

## ACHIEVEMENTS & CERTIFICATIONS

---

- **Top 4** — FASTATHON (SRM University All-India Hackathon)
- **Top 4** — Space Race Hackathon, Equinox Club, PES University
- [Adobe India Hackathon](#) — participant
- [Oracle Cloud Infrastructure](#) — AI Foundations Associate
- HackerRank — Problem Solving (Intermediate)