P Sri Ramaanuja Dasan

AI/ML Engineering Intern | B.E. Computer Science and Engineering (2nd year)

Chennai | linkedin.com/in/sri-ramaanuja-dasan | github.com/SomeNobody21112

PROFILE

Second-year CSE student (CGPA 8.03) who builds and ships AI systems end to end. Sole developer of **RAWRS**, a vision-language (Qwen2.5-VL) document accessibility platform that WinVinaya's remediators use almost every day to turn PDFs into accessible DOCX for blind and low-vision readers; it cuts remediation of a document from ~3 hours to 10-20 minutes. Open-source contributor to **llama.cpp** (merged PR adding GGUF quantization support). Tests design decisions against benchmarks and held-out test sets. Available for a 1-month internship (duration flexible), remote or on-site.

EXPERIENCE

Software Engineer Intern - AI Document Intelligence & Accessibility | WinVinaya InfoSystems, Bangalore — June 2026 - Present

- Build RAWRS (Remediation & Accessibility Work Reduction System) as sole developer: Qwen2.5-VL alt-text generation, table detection and remediation, page numbering and DOCX preview. Used almost daily by remediators producing accessible documents for blind institutions and colleges, one of the company's core services; cuts a document from ~3 hours to 10-20 minutes depending on page count and laptop hardware.
- Cut reviewer workload ~70% (from ~550 validation flags to ~165 real review actions) by finding ~385 duplicate checks and merging them, without deleting any evidence.
- Found three overlapping "is this document ready?" checks where the team assumed two, and merged them into one source of truth, so no fourth one was built.
- In progress: designing an automated pipeline (classify, interpret, fix, verify, review, export) in which every automated change stays reviewable and can be undone.

OPEN SOURCE

llama.cpp (ggml-org) | Contributor | PR #30630 — Oct 2026

- Added NumPy Q1_0 and Q2_0 quantization and dequantization to the gguf-py library, so Python tools can load 1-bit and 2-bit GGUF models (e.g. Bonsai) that previously failed with NotImplementedError. Matched the C reference bit for bit (66/66 checks across 27 quant types). Merged after maintainer review.

PROJECTS

RAWRS | Sole developer, WinVinaya InfoSystems — GitHub | rawrs-v1.vercel.app

Python, FastAPI, Pydantic, React, TypeScript, Qwen2.5-VL, Docling, PyMuPDF, Surya OCR, Mathpix, python-docx, pytest

- Wrote a 1,296-test suite and a 10-document benchmark that every architectural change must pass; it rejected several plausible changes, including a list detector that produced only false positives.
- Built one internal document model from which WCAG-compliant Markdown and DOCX are generated, so every reviewer decision is traceable and can be undone; an accessibility scorer reports blocking failures and export readiness.
- Tracks remaining failures by cause on the benchmark (e.g. missing H1 on 6 of 10 documents) to decide what to fix next, keeping human review on judgement-heavy decisions.

Thadam - Audit Intelligence for the MPLAD Scheme | Team lead, SIH 2026 (PS SIH26102) — GitHub

Python, pandas, web scraping, Sentence Transformers (MiniLM), scikit-learn, lifelines, FastAPI, React, Recharts, Agentforce

- Led a team of 6: assigned UI/UX and presentation work, frontend, and backend/ML by each member's skills, planned the architecture before coding, and ran every change through GitHub with logged reviews. Built the web-scraped data pipeline, the backend and the Agentforce integration.
- Analyses 987 public works and 476,781 lifecycle events: MiniLM sentence embeddings, 50 learned project archetypes, kNN peer comparison, and Cox survival models for completion risk, combined into one risk score. Ranks audit leads by expected value under inspector-time and travel budgets.
- With no labelled data, measured detection on synthetic anomalies injected into real records: 69.2% overall (96.1% stalled works, 83.2% inflated amounts). Outputs leads for inspectors, never fraud verdicts. ~160 tests, role-based access (JWT), tamper-evident audit log.

ICHNOVA - Blind Signal Analysis & Decoding | Sole developer, SIH 2026 (PS SIH26147) — GitHub

Python, NumPy, SciPy, React 19, TypeScript, Vite, Docker

- Built a receiver that decodes a raw radio capture with no known settings: it estimates carrier offset, symbol rate, modulation (BPSK/QPSK) and phase, then tests every error-correcting code and interleaver combination, and returns a result only if it is statistically significant. Implemented from first principles (Viterbi decoding, cumulants); held-out benchmark 30/30 with 0 false accepts.
- Validated on real over-the-air signals: decoded the WWV, WWVB, DCF77, MSF and JJY time signals (within 2-23 ms of GPS time) and the DDH47 teleprinter. Docker deployment with CI.

Other: AgriSense (voice-first farming advisory; presented at E-Summit, IIT Madras) | TruthTrace AI (misinformation verification) | Krishi Sakhi (marketplace with CNN-based produce quality assessment)

ACHIEVEMENTS

- Winner, SJIT internal Smart India Hackathon selection round (2025 and 2026)
- 3rd Place, Quantathon (SRM University) | 3rd Place, Cyberfest '26 (KCG College of Technology) | Special Mention, V-VORTEX (VIT Chennai)

SKILLS

AI / Machine Learning: LLM integration, vision-language models (Qwen2.5-VL), computer vision for documents (layout, table detection, OCR), NLP with sentence embeddings, clustering, anomaly detection, survival analysis, kNN, scikit-learn, model quantization (GGUF), agentic development with Claude Code
Languages & Frameworks: Python, TypeScript, JavaScript, React, Next.js, FastAPI, Pydantic, Tailwind CSS, pandas, NumPy, SciPy
Document & Accessibility: Docling, PyMuPDF, Surya OCR, Mathpix, python-docx, OOXML, WCAG 2.x | **Tools:** Git, GitHub, Docker, CI, pytest, Firebase, Vercel, web scraping

EDUCATION

St. Joseph's Institute of Technology (SJIT), Chennai | B.E. CSE (2nd year) — 2025 - 2029 | CGPA 8.03
Certifications: Data Science with Python | Fundamentals of Data Analytics | AWS Solutions Architecture (Job Simulation)