

Puripanda Venkata Praneeth

vpuripanda0506@gmail.com | 7019709567 | Bengaluru, India | github.com/imhunterr

SUMMARY

Backend engineer focused on distributed systems, database concurrency, and financial infrastructure. Built event driven, scalable, correctness-critical systems using Python, C++, PostgreSQL, RabbitMQ, Redis, and Docker.

EDUCATION

MIT Manipal

B.Tech in Mathematics and Computing

Manipal, India

2023 – 2027

EXPERIENCE

Celveza

Backend Engineering Intern

Hyderabad, India

May 2026 – Jul 2026

- Engineered **ExpressLane** , a 6-service event-driven open-road tolling backend simulating real-time RFID tolling, deployed across 12 Dockerized containers behind an Nginx gateway.
- Architected the scan-to-settlement path over 4 RabbitMQ queues with Dead-Letter Queues (DLQ), integrating a Redis sliding-window deduplication layer to prevent double-charging duplicate RFID reads.
- Guaranteed transactional consistency for wallet deductions using PostgreSQL row-level locks (SELECT FOR UPDATE), eliminating race conditions across an immutable 7-table audit ledger.
- Instrumented the stack with Prometheus and Grafana dashboards tracking lane throughput, queue depth, and failed-transaction rates under a sustained load of 100 scans/sec.

PROJECTS

LedgerMesh - Payment Reconciliation Engine | *Python, FastAPI, PostgreSQL, Gemini, Docker* 2026

- Architected a 5-layer reconciliation engine across 3 async financial streams, automating **455/495 orders (91.9%)** via deterministic SQL, contract formulas, and verified resolution patterns, with DB-enforced double-entry integrity.
- Built an earned-authority pattern pipeline (**candidate** → **shadow** → **approved** → **revoked**), promoting recurring resolutions only after independent verification and eliminating ~40 LLM calls per run.
- Hardened the system with **24 invariants and adversarial containment: 500/500 ground-truth accuracy**, while a bluff investigator had **60/60 resolutions blocked** with **Rs. 0 reaching the ledger**.
- Built a grounded finance console spanning reconciliation, cash forecasting, tax matching, human exceptions, and read-only SQL Q&A, with end-to-end attribution across deterministic and AI resolution paths.

x86-64 JIT Compiler | *C++17, x86-64, Linux* 2026

- Built a tiered JIT compiler and bytecode virtual machine from scratch in C++17, implementing a hand-written parser, stack-based bytecode IR, verifier, optimizer, interpreter, and direct x86-64 machine-code generation without external compiler libraries.
- Implemented On-Stack Replacement (OSR) and speculative deoptimization, transferring actively running interpreter frames into compiled machine code mid-loop and safely falling back to interpretation when runtime profiling assumptions break.
- Developed register allocation, CFG optimizations, and hot-loop profiling to achieve a 103× speedup guaranteeing memory safety via a 23-suite differential testing pipeline with 51,000+ fuzzed chunks and ASan/UBSan.

TECHNICAL SKILLS

Languages: C++, Python, SQL

Backend: FastAPI, REST

Databases: PostgreSQL, Redis

Messaging and Infra: RabbitMQ, Docker, Nginx