Suraj Van Verma

+1 (514) 641-6429 | vermasurajvan@gmail.com | linkedin.com/in/bythebug | github.com/bythebug | Montreal, QC, Canada

Skills

Languages: Python, Java, JavaScript, SQL
Frameworks & Libraries: FastAPI, Spring Boot, Django, Flask, SQLAlchemy, Pydantic | PostgreSQL, Redis, MongoDB
Cloud & DevOps: AWS (Lambda, S3, EC2, IAM), Docker, Linux, Git, CI/CD | Apache Kafka, RabbitMQ
Testing & Process: pytest, Unit Testing, Integration Testing, Code Review, Agile, Scrum
Architecture: RESTful APIs, Microservices, Distributed Systems, System Design, Event-Driven, WebSockets, JWT, RBAC
AI / ML: RAG, FAISS, LLM Integration (OpenAI, Ollama, Claude), NLP, Scikit-learn

Experience

Software Engineer Jun 2025 – Mar 2026
Plumfind *Montreal, QC*

- Architected an event-driven ingestion pipeline on **AWS Lambda** for NASA HLS satellite imagery; chose Lambda over persistent compute to handle irregular, burst-heavy data availability windows, processed ~**10,000 files** into NASA's Prithivi crop classification model schema and cut training prep from ~**3 days to under 2 hours**.
- Built a **Python** backend microservice that detects satellite data gaps and retries by expanding the time range then relaxing cloud-cover thresholds to maximize retrieval of clean, usable imagery; exposed as a **CLI tool** eliminating ad-hoc engineering support requests.
- Deployed a **pytest**-tested **RAG** pipeline indexing **1,000+** internal documents with **FAISS** vector search; Streamlit interface cut average knowledge lookup from ~**5 minutes to under 15 seconds** company-wide.

Software Engineer Sep 2023 – Jun 2025
McGill University, School of Computer Science *Montreal, QC*

- Designed a 3-layer **Spring Boot** backend (API, Application, Data) with independent modules (modeling engine, collaboration, workspace, grading); new modeling languages integrate via **EMF/Ecore** without touching core code, validated across **2 languages** in classroom deployments.
- Implemented real-time multi-user collaboration using **WebSockets** with **STOMP** protocol and **CRDTs** (Conflict-Free Replicated Data Types) for lock-free conflict resolution; supporting **20+ concurrent editors** with guaranteed consistency and presence tracking.
- Secured all endpoints with fine-grained **JWT** authentication and an **RBAC** permission matrix; implemented audit logging and **GDPR/FERPA**-compliant data handling, and integrated external LMS and automated grading services via RESTful APIs.

Software Developer Intern Aug 2022 – Nov 2022
Mitacs *Edmonton, AB*

- Delivered a conversational AI builder with a gamified interface enabling students to design and deploy chatbots hands-on; implemented **RBAC** for role separation and a **real-time chatroom** for in-session peer collaboration.

Education

McGill University Montreal, QC
Master of Science, Computer Science *Sep 2023 – Jun 2025*

- **Thesis:** *A Modular Backend for a Collaborative Educational Modeling Tool*
- **Coursework:** Applied Machine Learning | Natural Language Processing | Reinforcement Learning

VelTech University Chennai, India
Bachelor of Technology, Computer Science *Aug 2019 – Jun 2023*

- **Leadership: Google Developer Student Club (GDSC) Lead**; grew club membership to **2,000+** members and hosted **5+** technical events on backend development, ML, and open source contribution.
- **Coursework:** Data Structures & Algorithms | Operating Systems | Database Management Systems

Projects

urltrim | *FastAPI, PostgreSQL, Redis, RabbitMQ* `github.com/bythebug/urltrim`

- Async Python backend with a **Redis** caching layer serving repeat URLs at sub-millisecond latency and **RabbitMQ** decoupling click-event processing from the request path; benchmarked at ~**3,000 req/s** on a single node.

Real-Time Distributed Log Processing System | *Kafka, Redis, Docker, Python* `github.com/bythebug/RT-Log-Processor`

- **Kafka** consumers fan out log events to parallel processing workers handling **500K+ events/sec**; **Redis** caches recent log state for sub-millisecond lookups; achieved **40% latency reduction** under peak load.

context-os | *Python, RAG, LLMs (OpenAI, Ollama, Claude)* `github.com/bythebug/context-os`

- Production-grade multi-tenant memory API for **LLM** apps; hybrid **BM25** + **pgvector** retrieval fused via **RRF**, **Redis** caching for sub-60ms repeat queries, per-key rate limiting, and **Python** + **TypeScript** SDKs; deployed on Fly.io.