

REX REMIGIUS STEPHEN JOTHI

+1 (857) 396-3061 | ✉ rexremigius1@gmail.com | [in linkedin](#) | [🌐 portfolio](#) | [github](#)

EDUCATION

Northeastern University, Boston, MA Master of Science in Computer Science Relevant Course Work : Programming Design Paradigm, Computer Systems	2025 - Expected 2027 (GPA: 3.834 / 4.00)
Sri Krishna College of Engineering and Technology, Coimbatore, India Bachelor of Technology in Information Technology Relevant Course Work: Object Oriented Programming, Computer Architecture, Data Structures and Algorithms, Machine Learning Techniques and Big Data Analytics	2019 - 2023 (GPA: 8.68 / 10.00)

SKILLS

Programming Languages:	C, C++, Python, Java
Compiler & Systems:	LLVM, GCC, Intrinsic Programming, SIMD Optimization
Machine Learning & AI Frameworks:	TensorFlow, PyTorch, ONNX, TFLite, NumPy, Image Processing
Development & Testing Tools:	Git/GitHub, Make, Linux, CMake, PyTest, LIT, Docker, VSCode, Jupyter

WORK EXPERIENCE

Software Engineer, MulticoreWare Inc	Jan 2023 - Jul 2025
<ul style="list-style-type: none">Improved compiler performance and maintainability by designing custom intrinsics and optimization passes in LLVM and GCC, enabling reusable target-specific code and improving generated code efficiency by 15% through memory-aware transformations.Reduced inference latency on a custom AI accelerator by architecting and optimizing ML kernels (Convolution, Maxpool, Normalization), applying SIMD optimization and cache-efficient memory layouts to achieve 63% single-stream and 67% multi-stream latency reduction.Enabled efficient deployment of large ML models on specialized hardware by porting workloads from ARM CPU to Cadence Vision P6 DSP using the Xtensa simulator, incorporating memory modeling to validate theoretical estimates and guide kernel optimization.Streamlined model deployment pipelines by converting PyTorch models to ONNX and fully quantized TFLite, building automated verification workflows to ensure bit-exact output consistency and low-latency inference across platforms.	
Summer Intern, Zoho Corporation	Apr 2022 - May 2022
<ul style="list-style-type: none">Strengthened core systems and OOP fundamentals by implementing Java-based CLI applications using collections and data structures, improving problem-solving efficiency through structured debugging and algorithmic reasoning.Applied competitive programming techniques to optimize logic and improve code correctness, gaining hands-on experience with performance trade-offs and clean API design.	

PROJECTS

Enhanced Address Sanitizer pass in LLVM for fault tolerance system	C, LLVM, C++, Shell, Make
<ul style="list-style-type: none">Addressed program crashes caused by memory errors by extending LLVM's AddressSanitizer to support fault-tolerant execution instead of aborting, implementing the CIMA(Countering Illegal Memory Access) algorithm with CFG-based instrumentation.Improved runtime availability and diagnostics by integrating custom LLVM/Clang passes, enhanced reporting, and adaptive memory resizing, achieving 100% runtime availability, 0% crash rate, and 30% lower reallocation overhead, validated using IR-level testing and C benchmarks.	
Transparent Checkpoint and Restart System	C, POSIX signals, Linux syscalls, GDB
<ul style="list-style-type: none">Built a transparent checkpoint-restart mechanism to enable fault tolerance and process recovery by capturing full process state (memory segments and CPU registers) using /proc/self/maps and getcontext/setcontext.Ensured accurate process restoration by reconstructing execution in a new address space without modifying target programs, validating correctness through controlled restart experiments and state consistency checks.	
Heart Disease Risk Prediction Model	Python, Pandas, NumPy, Matplotlib, Scikit-learn
<ul style="list-style-type: none">Improved predictive accuracy on structured medical data by building and tuning a logistic regression model on the UCI Heart Disease dataset using feature engineering, preprocessing, and hyperparameter optimization and achieved robust model performance with 90% accuracy, validated using precision, recall, and ROC-AUC metrics to ensure generalization and reliability.	