Anthony Nwafor

LinkedIn ❖ (662) 457-0162 ❖ anthonynwafor261@gmail.com ❖ https://7bbg.github.io/portfolio/

SKILLS

- **Programming Languages:** C/C++, JavaScript, Java, HTML, CSS, SQL, Python, Typescript,
- **Frameworks:** TensorFlow, Keras, Sci-kit-learn, NumPy, Pandas, React, Flask, Django, Pytorch, ElectronJs, CVXOPT, PyQT, CUDA, vLLM.
- **Technologies:** GitHub, Git, MATLAB, Linux, MYSQL, NOSQL,

EDUCATION

Mississippi Valley State University **May 2026**

Bachelor of Science in Computer Science & Mathematics

Relevant Coursework: Data Structures and Algorithms, Operating Systems, Software Engineering, Database Management, Discrete Structures, Language & Compiler, Introduction to Networking.

WORK EXPERIENCE

ActiveCampaign, Software Engineering Intern | Summer 2024, Indianapolis, IN

Developed ActiveCampaign contact tags via Zapier and automated scripts via a CLI tool.

- Worked on a flexible Zapier action to update contact tags based on event triggers, even when only the email address is provided.
- Developed a command-line tool (CLI) to manage and automate the execution of custom scripts within the ActiveCampaign Automation Platform.

Science Gateway Institute, Software Engineering Intern | Summer 2023, Itta Bena, MS

Developed an NLP chatbot web application, deployed to AWS, increasing user traffic.

- Co-developed a full-stack web application with a natural language processing (NLP) chatbot using Python, cloud computing (AWS), and full-stack technologies (HTML, CSS, Figma, JavaScript, SQL, PHP, and cPanel).
- Utilized Python to develop the NLP chatbot, AWS to host the application, and full-stack technologies to build the user interface.

SELECTED PROJECTS

FlashAttention CUDA Kernel: High-Performance LLM Acceleration. |Github Code , 2026

An implementation of CUDA C++ FlashAttention implementation optimizing HBM-to-SRAM data movement to bypass the GPU memory wall for accelerated Transformer training and inference.

- Engineered a custom CUDA C++ FlashAttention kernel with an autograd-compatible backward pass, utilizing Online Softmax and on-the-fly recomputation to reduce memory complexity from $O(N^2)$ to $O(N)$ while maintaining 1e-5 numerical precision.
- Optimized the GPU memory hierarchy via SRAM tiling and cp.async (Ampere/Hopper) to overlap Tensor Core computation with HBM data movement, achieving a functional 1.31 TFLOPS baseline for high-performance Transformer training and inference

LSM-DB: Log-Structured Merge-Tree Storage Engine | Github Code, 2025

Engineered an LSM-Tree C++ storage engine featuring a SkipList MemTable, WAL, Bloom Filters, Sparse Index, and Compaction Engine for high-performance, durable NoSQL data storage.

- Engineered "LSM-DB," a core database storage engine in C++ implementing the Log-Structured Merge-Tree architecture, a fundamental design for high-write-throughput NoSQL databases like Cassandra and RocksDB.
- Engineered a durable and performant data persistence system, featuring a SkipList-backed MemTable for in-memory writes, a Write-Ahead Log (WAL) for crash recovery, and efficient SSTables on disk.
- Designed and implemented I/O optimization algorithms, including Bloom Filters for fast negative lookups and a Sparse Index for efficient disk access, significantly accelerating the read path.
- Managed data lifecycle using a multi-component system, incorporating a background Compaction Engine that performs K-way merge sort to consolidate SSTables, reclaim space, and maintain only the latest key versions.

Activities and Awards

Presidential Scholar , Peer Tutor at CS & Math Department at MVSU