

OLADELE UTHMAN

uoladele99@gmail.com • +234 706 343 2968 • github.com/codetesla51 • devuthman.vercel.app

PROFILE

Backend engineer specializing in systems programming, concurrency, and performance optimization. Builds production-grade backend systems from first principles using Go and PHP. Focused on identifying bottlenecks, profiling code, and delivering measurable performance improvements.

TECHNICAL SKILLS

Languages	Go, PHP, JavaScript, Bash
Databases	PostgreSQL, MySQL, Redis
Tools	Docker, Git
Core Work	Parser design, protocol implementation, concurrent systems, performance optimization

EXPERIENCE

Independent Backend Engineer 2023 — Present

- Design and implement production-grade backend systems in Go and PHP, emphasizing correctness and performance
- Profile and optimize networked services to identify bottlenecks; delivered 16x throughput improvement through socket optimization and connection pooling
- Develop parser tooling and authentication libraries with test-driven development; maintain >80% code coverage

FEATURED PROJECTS

Axion [GitHub]

CLI mathematics engine with recursive descent parser and AST evaluation

- Scientific functions, unit conversion, persistent history
- Demonstrates parser design and compiler fundamentals

Go • Parser Design • AST • Testing

Raw-HTTP [GitHub]

HTTP/1.1 server built on TCP sockets without frameworks

- Implemented custom request parser, persistent connections, connection pooling
- Achieved 16x performance gain: 250 RPS → 4,000 RPS through bottleneck analysis
- Demonstrates protocol implementation and socket-level optimization

Go • TCP Sockets • HTTP Protocol • Performance Profiling

GO-CHAT [GitHub]

Real-time concurrent chat server with 100+ concurrent user capacity

- Goroutine-based concurrency with channel message dispatch
- TLS encryption and rate limiting for production use
- Demonstrates safe concurrent system design

Go • TCP Networking • TLS • Concurrency

GoLexer

[GitHub]

High-performance tokenization library for compiler tooling

- 50+ token types with full UTF-8 handling
- Validated against 1,700+ token edge cases
- Compiler infrastructure component

Go • Lexical Analysis • Compiler Tools

Mini-Git

In Progress

Re-implementing core Git primitives to understand version control internals

- Content-addressable storage with SHA-256 hashing
- Blob storage, tree structures, commit graphs
- zlib compression and binary-safe serialization

Go • Filesystems • Git Internals

EDUCATION

B.Sc. Computer Science — University of Ilorin (In Progress)