
JACOB LANE MOORE

jlmoors001@gmail.com ◇ (256) 701-3403 ◇ github.com/jmogainz ◇ linkedin.com/in/jacob-moore-3347b519b

EDUCATION

MS, Georgia Institute of Technology, Computer Science, Computing Systems Expected 2027

- GPA: 4.0/4.0
- Coursework: Hypervisors, Distributed Systems, Software Architecture, Compilers

BS, Mississippi State University, Computer Engineering Graduated

- GPA: 4.0/4.0
- Coursework: Embedded Systems, Systems Programming, Computer Architecture, Operating Systems

ABOUT ME

Languages	Golang, C++/C, Python, Typescript, Java, Dart
Development	Linux, Git, Docker, PostgreSQL, BigQuery, Databricks, Neovim, Claude Code, Next.js/React
Certs	CompTIA Security+, Neural Networks and Deep Learning (DeepLearning.AI, Andrew Ng)

EXPERIENCE

Software Engineer, Walmart 03/2026 – Present

- Contributed to an internal analytics dashboard MVP using BigQuery, Cube.js, MCP-backed services, and a LangGraph orchestration layer; worked across data modeling, backend services, orchestration, and UI.
- Raised a distributed microservices system from ~30 to ~300 bidirectional TPS by instrumenting critical paths, identifying CPU- and I/O-bound hotspots, and eliminating code, network, and concurrency bottlenecks.
- Own backend features end to end, orchestrating 4–5 concurrent AI-agent workstreams to accelerate stakeholder delivery across Go REST APIs, Kafka/NATS event-driven services, and technical-debt improvements.
- Standardized Go-service observability and reliability with Prometheus metrics, structured logging/error standards, OpenObserve alerts, automated migrations, and layered E2E/CI tests.
- Built and maintain a Kubernetes operations console adopted by the team for logs/metrics, pod actions, environment drift, and CI/CD deployment state.
- Built and maintain a CI shift-left CLI used by multiple teams to compile heterogeneous stacks, parallelize Sonar/Snyk/secret scans in isolated Git workspaces, enforce CI gates, and guide agents/developers through remediation.

Software Engineer, Torch Technologies 05/2023 – 03/2026

- Maintained and extended a logically clocked C++ microservices platform powering distributed RL training backends and real-time Unreal Engine visualization.
- Designed an RL observations DSL and plugin architecture enabling schema-driven data extraction and configuration-based swapping of actions, observations, and rewards without code changes.
- Integrated C++ simulation environments with RLlib/SB3 via pybind11 and orchestrated isolated, vectorized Docker environments for high-throughput synchronized training.
- Developed high-performance C++ data models and ONNX Runtime integrations for large-scale classification and forecasting datasets, model evaluation, and real-time inference across neural-network architectures.
- Contributed to a LangGraph-based internal engineering agent using open-weight models to answer questions over restricted codebases and generate plugins and simulation scenario configurations.
- Built a multi-threaded observability framework across microservices with time-series and event capture, enabling distributed debugging and performance analysis.

Startup Founder (CTO), Poof LLC ◇ thepoofapp.com

10/2024 – 01/2026

- Built and shipped a 100,000+ LOC gig-work platform: Go/PostgreSQL microservices, Flutter worker apps on  iOS ◇  Android, and customer web dashboards.
- Automated local-to-production workflows with Make/Docker Compose and blue-green CI/CD across Dev/Staging/Prod on Fly.io behind Cloudflare, integrating Bitwarden secrets and layered end-to-end/integration tests.
- Built secure auth with JWT (HttpOnly cookies, short-lived access, refresh flow) and device attestation (App Attest, Play Integrity) for high-trust actions.
- Built idempotent, failure-safe webhook workflows for Stripe Connect payouts, Checkr background checks, and Stripe Identity verification.
- Built job scheduling/assurance with surge pricing, worker reliability scoring, GPS-tracking, and AI photo verification to guarantee fulfillment.

Software Engineering Intern, Torch Technologies

05/2020 – 08/2022

(Summers/Winters)

- Built Python/C++ data pipelines within a distributed microservices system, running thousands of simulation scenarios to generate millions of samples for neural-network training and evaluation.
- Trained and evaluated CNN, RNN/LSTM, and transformer architectures alongside classical models for classification and time-series forecasting.
- Extended Unreal Engine C++ front-end and tooling for interactive, distributed simulation visualization.

Undergraduate Research Assistant, Mississippi State University CSE Department

09/2020 – 05/2022

- Researched predictive maintenance using deep learning and built data acquisition pipelines interfacing with fiber optic sensing hardware.

SKILLS

- **Software Architecture & Design:** System design patterns (OOP), modular and plugin architectures, service-oriented architectures, event-driven systems, API design, performance-critical system design.
- **Backend & DevOps:** Go microservices, PostgreSQL, containerization (Docker), CI/CD pipelines (GitHub Actions), secrets management (Vault/Bitwarden), feature flags (LaunchDarkly), automation tooling (Make/Bash).
- **Systems Programming:** C/C++, POSIX APIs, concurrency primitives, networking stacks (TCP/UDP), CMake/Make, performance profiling, cache utilization.
- **Distributed Systems:** Network protocols (gRPC/REST, custom UDP, FlatBuffers), messaging/streaming (Kafka, RabbitMQ, NATS), distributed data systems, asynchronous communication, logical clocking, publish-subscribe patterns.
- **ML Integration:** PyTorch/TensorFlow, ONNX Runtime; CNN, RNN/LSTM, and transformer architectures; classification, time-series forecasting, and RL pipelines (Stable Baselines, RLlib).

PROJECTS

mazle.io – Daily Puzzle Game ◇ mazle.io

- Built a daily Pokémon-inspired maze puzzle game with 100+ DAU featuring ice sliding mechanics, one-way ledges, and step movement.
- Designed and implemented DL/RL-based maze generation algorithm for creating psychologically challenging puzzles with counter-intuitive optimal paths and deception patterns.
- Engineered full-stack system: React (Next.js)/TypeScript frontend with Phaser game engine, Rust backend with WASM fallback, automated daily puzzle generation via Vercel Cron with KV caching.

Compiler

- Designed and implemented a C++ compiler (~16k LOC) for an ALGOL-like language, producing optimized MIPS assembly.
- Implemented semantic analysis, IR optimization, and register allocation (Chaitin naive, Briggs optimization) with CFG and liveness analysis.
- Achieved best-in-class performance at 10x lower compute cost compared to CS PhD peers.

tryhumansonly.com – Cognitive Benchmark Platform ◇ tryhumansonly.com

- Built a minimalist cognitive benchmarking platform with 18 precision tests (reaction time, memory, logic, spatial reasoning, and more), global leaderboards, and user accounts.

Before and After Creator (Car Detailing Clustering Website)

- Built a web app using unsupervised ML/DL to automatically cluster and pair before/after car detailing photos for a local car detailing business.
- Reduced manual photo matching and collage creation time from 30 minutes to seconds for client deliverables and advertising materials.