SAMUEL DARIUS

Holbrook, MA 02343 | samueldarius1470@gmail.com
linkedin.com/in/samdarius | github.com/Kingsam147 | samuel-s-portfolio.vercel.app

SUMMARY

Backend engineer who leads with system design — data modeling from access patterns, multi-tier caching, multi-tenant authorization, and CI/CD, with Domain-Driven Design as a differentiator. Makes architecture decisions on measured speed, performance, and cost outcomes.

TECHNICAL SKILLS

Languages: JavaScript/TypeScript (primary), Python, SQL, Java, C

Backend: Node.js, Express, REST API design, serverless architecture, webhook integrations

Databases & Caching: MongoDB, PostgreSQL, MySQL, Redis, localStorage caching strategy

Cloud & DevOps: AWS (S3, EC2), Docker, Vercel, GitHub Actions (CI/CD), Cloudflare (CDN, WAF, rate limiting)

Auth & Security: Auth0, OAuth 2.0, JWT validation, per-user authorization, HMAC webhook verification

Architecture & AI: Domain-Driven Design (layered architecture, entities, repositories), optimistic locking, LLM cost-based model routing, agent context-window caching

EXPERIENCE

TourGate — Co-Founder & Technical Lead *2025 – Present*

- Own all technical decisions for an early-stage startup that unifies real estate agent messages from Facebook, Apartments.com, Craigslist, Redfin, MLS, and YGL into one inbox with Google Calendar integration; refactored and security-hardened inherited starter code.
- Built an autonomous AI agent that reads and replies to messages on agents' behalf with per-user configurable settings.
- Implemented cost-based LLM routing — simple messages to a cheaper model, complex ones to a more capable model — reducing API spend without degrading response quality.
- Engineered a Redis sliding-window cache (last 20 messages/user) for the AI context window to bound memory and keep responses fast; chose PostgreSQL (Supabase) over MongoDB as the system of record after modeling the data as relational with narrow-field queries.

Software Engineer Fellow — Hack.Diversity, Boston, MA *Jan 2025 – May 2025*

- Took ownership of an inherited, error-ridden Pinterest clone — traced an unfamiliar codebase, diagnosed the failures across the application, and debugged it into a working state.
- Solved 18+ algorithm problems (dynamic programming, recursion, hash maps) applying time/space complexity optimization.
- Practiced production Git workflows: branching, pull requests, code review, and Agile collaboration.

PROJECTS

Pokémon Radical Red Companion & Damage Calculator (Independent Full-Stack) *Live Demo · GitHub*

- Built the only companion tool for Pokémon Radical Red, a fan-made ROM hack with custom mechanics no public API covers — players track a playthrough by storing the Pokémon they've caught and running damage calculations unavailable anywhere else.
- Extracted and structured the full game dataset from the ROM's source code (1,200+ Pokémon, 900+ moves, 300+ abilities), served over 32 REST endpoints on a Domain-Driven Design backend.
- Chose MongoDB over SQL for deeply nested objects always fetched whole — cutting data load times 5–10x; isolated per-user data with multi-tenant Auth0 authorization checked against resource ownership.
- Cut enemy-team response time from 1,384ms to 196ms (7x) with shared Redis — the only viable cache in a serverless backend; lazy-loaded only the active box to cut initial load and memory.
- Shut down a cost spike from scraping traffic hammering an exposed EC2 endpoint (~$60 bill) by adding Cloudflare rate limiting — returning monthly AWS spend to ~$0.38.

Car Detailing Booking Platform (Independent) *Live Demo · GitHub*

- Replaced phone bookings with a self-serve system that computes all 19 daily time slots on the fly and subtracts booked ones — no slot table to maintain; O(1) availability checks via a Set.
- Lazy-loads slots only on date selection (zero DB hits from non-booking visitors) behind a 5-minute Cloudflare edge cache; duplicate bookings rejected at write time.

Student Scheduling System (Scrum Master · 5-developer Agile team) *Live Demo · GitHub*

- Led a 2–3 month cycle (sprint planning, stand-ups, ticket tracking); assigned work by skill level to protect sprint throughput and keep the team unblocked.
- Built Google OAuth with server-side ID token verification and backend validation cross-checking availability against uploaded academic schedules (ICS), eliminating conflicts in a client's manual, spreadsheet-based workflow.

EDUCATION

University of Massachusetts Boston — B.A. Computer Science (Honors) *GPA 3.4*

ACTIVITIES

Cross Country & Track — UMass Boston (2023–24) | Church Volunteer — Beth Schilo & PCLH (2018–Present, 8 yrs)