JARED MAHOTIERE

Jared Mahotiere | Bear, DE | (302) 803-7673 | jmahotie@purdue.edu | linkedin.com/in/jared-mahotiere | github.com/jmahotiedu | jmahotiedu.github.io/jmahotiedu

SUMMARY

Backend and systems software engineer focused on distributed services, C systems programming, .NET backend development, SQL performance optimization, AWS ECS deployments, and open-source architecture hardening.

EDUCATION

Purdue University - B.S. Electrical Engineering Technology (Computer Engineering Technology)
Minor: Computer & IT | Certificate: Entrepreneurship & Innovation | Expected May 2026

LEADERSHIP & ORGANIZATIONS

Delta Tau Delta (Campus Chapter): DEI Chair | **National Society of Black Engineers (NSBE):** Member

SKILLS

Languages: C, C#, TypeScript, Python, Java | Backend: .NET, ASP.NET Core, Node.js, Express, REST, gRPC, concurrency, POSIX networking | Data: PySpark, Kafka, XGBoost, Prophet, FastAPI | Storage/Infra: PostgreSQL, SQL Server, Redis Streams, Docker, GitHub Actions, Prometheus, Grafana, AWS (ECS, ALB, RDS, ElastiCache), Terraform

EXPERIENCE

Nucor Corporation - Software/Automation Engineering Intern | Darlington, SC | May-Aug 2024 and May-Aug 2025

- Developed and maintained Blazor/.NET real-time operator dashboards and robust back-end services, enhancing process transparency and improving steel production workflows.
- Managed and analyzed production data in SQL Server/QMOS databases; developed optimized queries and recommended new tables/columns to support process improvement.
- Built automated reporting and alert systems using Quartz.NET with real-time email notifications for maintenance and quality events, reducing manual monitoring and accelerating response.
- Migrated legacy Visual Basic applications to .NET/Blazor, reducing technical debt; utilized Git for version control, peer code reviews, and codebase integrity.

PROJECTS

workflow-orchestrator - TypeScript, Node.js, Redis Streams, Postgres

- Built a distributed workflow engine with DAG validation, Redis Streams consumer groups, idempotent retries, and durable Postgres run/task state; benchmarked 25/25 runs in 15.94s and deployed on AWS ECS Fargate with ALB/RDS/Redis.

IoT Streaming ETL Pipeline - Kafka, PySpark, Airflow, Redshift

- Implemented event-driven ingestion at 100+ events/sec with medallion data architecture, quality validation, and production monitoring/alerting.

Retail Sales Forecasting Dashboard - Python, XGBoost, FastAPI, AWS ECS

- Shipped a live forecasting product on AWS ECS; achieved XGBoost R2=0.91 and delivered 90%+ automated test coverage for API/model workflows.

OPEN SOURCE CONTRIBUTIONS

- PicoClaw (Go): 3 merged PRs on a 14.5k-star project, including provider protocol-family refactor (#213) and security/model hardening follow-ups; invited to Dev Group.
- Bloomberg comdb2 (C/Java): fixed JDBC metadata cursor isolation bug in PR #5731 to preserve active getTables() result sets during version lookup.
- Bloomberg comdb2 (C/C++/SQL): backported targeted SQLite security fixes for issue #3904 in PR #5743 (commit cede68b52); built from source, ran full test harness, reproduced failures by test ID, and published a security-fix verification matrix.