Mohammad Traore

U.S. Citizen | 934-386-6497 | mohammadtmt23@gmail.com | linkedin.com/in/mohammadtmt | mohammadtmt23.github.io

Education

Syracuse University — Syracuse, NY
B.S. Mechanical Engineering, Minor in Computer Science — *Expected May 2026*

Relevant Coursework: Data Structures, Object-Oriented Programming, Language Design & Implementation, Systems & Network Programming, Algorithms, Electrical Engineering Fundamentals, Data Analysis for Engineers, Control Systems, Differential Equations

Experience

AI Fellow – Handshake — November 2025 – Present
Handshake Fellowship Program — *Remote*

- Selected for a competitive paid fellowship focused on real-world AI model development, evaluation, and deployment
- Engineered an automated evaluation pipeline in Python processing 10,000+ AI-generated outputs across multimodal tasks (text, audio, visual), improving model accuracy by 15%
- Built a structured data-labeling system for LLM fine-tuning, contributing annotated feedback datasets to Project Hedgehog and Project Lexicon
- Implemented a comprehensive testing framework and CI/CD pipelines ensuring reproducible, version-controlled AI evaluation workflows

Mechanical Design Engineer – JMA Wireless — September 2025 – Present
RF Radome Wind-Load Optimization — *Syracuse, NY*

- Designing parametric radome cross-sections and wind-tunnel test fixtures in SolidWorks targeting $\geq$10% wind-load reduction; producing manufacturing-ready drawings with GD&T callouts and safety factors $\geq$2
- Executing steady-state and transient CFD simulations in ANSYS Fluent with inflation-layer meshing and mesh-independence verification; predicting drag coefficients (C_d) validated against wind-tunnel data within $\leq$10% error
- Delivering weekly technical briefings to engineering leadership summarizing design progress, simulation results, and risk mitigation strategies

Systems & IT Support – Syracuse University Esports — January 2025 – Present
Syracuse University — *Syracuse, NY*

- Automated scheduling and system-configuration workflows using Python scripts, improving operational efficiency by 40%
- Developed event-management software integrating third-party APIs for real-time data reporting and decision support
- Diagnosed and resolved software and hardware issues in live production environments supporting 100+ concurrent users

Skills

Languages: Python, Java, C, JavaScript, MATLAB, Racket, Haskell, LaTeX
Frameworks & Libraries: TensorFlow, PyTorch, NumPy, Pandas, OpenCV, React, Node.js
Tools & Platforms: Git, GitHub, Docker, AWS, Linux, Jupyter, ANSYS, SolidWorks
Concepts: Data Structures & Algorithms, OOP, RESTful APIs, CI/CD, Machine Learning, Deep Learning, Agile

Projects

Self-Driving Car Simulation | *Python, TensorFlow, NumPy, OpenCV* — October 2025 – Present

- Built an end-to-end autonomous driving system in Python/TensorFlow, processing 50,000+ images from Udacity's simulator; engineered an ETL pipeline with data augmentation (cropping, normalization, flipping, class balancing) improving model generalization by 25%
- Designed and trained a CNN based on NVIDIA's end-to-end autonomous-driving architecture achieving 95% steering accuracy on the validation set; deployed model in closed-loop simulation with real-time inference at 30 FPS

Campus Bookstore Manager & Pointer Systems | *C* — Fall 2025

- Implemented a modular C program with do-while input validation, five decoupled functions (`readPrices`, `calcTotal`, `applyDiscount`, `findMaxIndex`, `countAboveAvg`), and a structured summary report demonstrating clean separation of concerns
- Built a five-part pointer deep-dive covering address inspection, in-place array modification via `const double *` and pointer arithmetic, `sizeof` memory audits, and a sales-summary engine using pure pointer notation `*(ptr+i)` — no array subscripts

Tic-Tac-Toe with MiniMax AI | *Racket, Functional Programming* — Fall 2025

- Implemented a full Tic-Tac-Toe engine in Racket supporting arbitrary N×N boards; wrote `board?`, `next-move`, `valid-move?`, `make-move`, and `winner?` using purely recursive functional patterns with no mutation
- Developed a bonus `calculate-next-move` AI using the MiniMax algorithm with a playable GUI runner (`gui.rkt`) supporting verbose mode, AI mode, and configurable board size via command-line flags