Nathan Samson
Baltimore, MD | 443-579-6803 | Natestesa@gmail.com | LinkedIn | GitHub | Personal Website

EDUCATION

University of Maryland Baltimore County Baltimore, MD
Bachelor of Computer Science Expected May 2026
Relevant Coursework: CMSC201(Python), CMSC201(C++), Calc 1, Calc 2, Discrete Math 3.5 GPA
Societies: EESA (Ethiopian Eritrean student association), NSBE, Software Development and Administration Club

TECHNICAL SKILLS

Languages: Python, C++, Java, JavaScript, Java, SQL, HTML, CSS
Technologies: TensorFlow, scikit-learn, Flask, TSDB, MQTT, Git, PyTorch, Linux OS, Pandas, React, Node.js
Concepts: Artificial Intelligence, Machine Learning, Natural Language Processing, Data Analysis, IoT, REST APIs, Data Management, Compiler Theory, Prompt engineering, Edge Computing

WORK EXPERIENCE

OmniSyncAI Remote
Full stack Software Engineer intern May 2024 – Now

- Currently working developing Account setup for the company. Implementing Business account setup, Ability to invite team member, 2FA verification, AI recommendation for business with Node.js, React, and PostgreSQL

Radical AI Remote
Artificial Intelligence Engineer intern Apr 2024 – Now

- As an Artificial Intelligence Engineer at Radical AI, I am leveraging technologies such as OpenAI and Google Gemini to develop ReX, an AI coach who serves as a steadfast career companion for learners, offering personalized coaching, mentorship, and support throughout the various phases of their career lifecycle.

UMBC Baltimore, MD
Research Assistant Aug 2023 – Now

- Collaborated within the DAMS research lab(https://damslabumbc.github.io/), specializing in IoT, where I actively engaged with IoT devices and systems while developing code to extract valuable insights from sensor data.
- Developed algorithms for IoT device data analysis, focusing on occupancy detection, and environmental monitoring.
- Designed server infrastructure on Linux cloud OS, managed databases (SQL and INFLUXDB), Utilized MQTT broker (EMQX) for communication between lidar sensor and implemented RESTful APIs for seamless data access.
- Leveraged the SARIMA (Seasonal AutoRegressive Integrated Moving Average) algorithm to predict future trends and patterns in sensor data, significantly enhancing the lab's capability to forecast occupancy levels and environmental conditions and inform proactive decision-making in IoT environments.
- Integrated edge computing to optimize server processing for real-time data handling from IoT sensors.

PROJECTS

YouTube Comments Extraction and Sentiment Analysis Project
Skills: Python, Prompt engineering, Natural Language Processing (NLP), Machine Learning, Data Preprocessing, API usage, Text Analytics

- A Python-based project, I leveraged the YouTube Data API to gather comments from videos, aiming to analyze the underlying sentiments. Utilizing advanced NLP techniques and sophisticated machine learning models facilitated by TensorFlow and scikit-learn, I categorized the comments based on their sentiments. The process was further refined through prompt engineering with the LLaMA Index, ensuring nuanced sentiment detection. This comprehensive approach not only accurately identified sentiment trends but also equipped content creators with deep insights into viewer reactions. The results were presented through an intuitively designed interface, significantly aiding in the strategic optimization of digital content.

SMART Campus
Skills: Python, SQL, IoT, Data Architecture, Machine learning, Data Structures, Algorithms, API development, InfluxDB, Flask, REST APIS, Real-Time Data Pipeline Development, HTTP Communication Protocols, EMQX, MQTT, Git

- Currently working on a project to make a Smart Campus. focusing on real-time space availability using IoT sensors. Responsibilities include database management, algorithm implementation, and API development, showcasing proficiency in Python, SQL, and data architecture.

Advanced Connect4 AI with Reinforcement Learning
Skills: Python, PyTorch, Reinforcement Learning, Neural Networks, Flask, API Development, Noisy Nets, Prioritized Experience Replay, Dueling Network Architecture

- In this project, I developed an AI for playing Connect4, employing advanced Reinforcement Learning techniques including Distributional Dueling Networks, Noisy Nets, and Prioritized Experience Replay, all built with PyTorch. The AI was integrated into a custom game environment and connected to a real-time interface via a Flask API. The setup, optimized for CUDA-enabled GPUs, allowed for efficient training and evaluation. Performance metrics were closely monitored using TensorBoard, facilitating fine-tuning of the neural network. Additionally, opponent modeling was implemented to enhance the AI's adaptability against various strategies, significantly improving its performance against both conventional algorithms and human players.