

Vaman Rajagopal

856-360-0338 | vaman@umich.edu | [linkedin.com/in/vamanr](https://www.linkedin.com/in/vamanr) | vamanr06.github.io

EDUCATION

University of Michigan

Ann Arbor, MI

BSE in Computer Science and Robotics

Anticipated Graduation Date: May 2028

- Courses: Data Structures and Algorithms, Operating Systems, Machine Learning, Agentic AI Software Engineering

EXPERIENCE

Software Developer

June 2026 – Present

Exar North

Remote

- Built a modular **React, TypeScript, and Supabase** architecture across **5 sports media websites**, reusing shared templates, components, data modules, and a common publishing workflow to launch and maintain new sites
- Implemented game-aware TTL caching for sports APIs, with modeled results of **approximately 43% fewer requests** and **3.3× higher live-game refresh frequency** than fixed polling
- Automated article production in **under 3 minutes** without manual writing or editing by combining web search, API-sourced images, and **Gemini and Claude** content-sufficiency checks across all 5 sites

Software Developer

Mar. 2025 – Present

Michigan Institute for Computational Discovery and Engineering

Ann Arbor, MI

- Engineered a stateful graph execution runtime in **React, TypeScript, and Python** that preserves conversation state and reusable variables across **OpenAI** calls, simulated loops, and semantic or deterministic branches
- Tested end-to-end conversational flows for response accuracy, tool behavior, and failure handling across **model calls, external APIs, extraction, loops, and branching** in the visual workflow platform
- Integrated **OpenAI function calling, vector retrieval, and external APIs** into node-based workflows with configurable controls for reasoning, looping, response evaluation, and multi-step tool execution

Software Engineer

Jan. 2026 – Present

Michigan Edge

Ann Arbor, MI

- Built a **Python, BeautifulSoup, and Selenium** data pipeline to ingest live betting lines across multiple sportsbooks and continuously normalize changing markets into a unified downstream data model
- Automated sportsbook navigation and bet placement with **Selenium**, reducing end-to-end placement time by **approximately 80%** for time-sensitive wagers while preserving selected markets and odds
- Normalized sportsbook-specific **team, market, and odds formats** into consistent records consumed by downstream automation without source-specific processing logic

Backend Developer

June 2025 – May 2026

Everstory

Ann Arbor, MI

- Built a **PostGIS**-backed geospatial data model for an audio-first encyclopedia with **100+ entries**, mapping content to geographic origins and supporting location-aware discovery through interactive maps
- Developed **FastAPI and Supabase** services for content, geographic metadata, and historical context, supporting voice-driven retrieval and interactive browsing across the **100+ entry** collection

PROJECTS

Voxyl Statistics | *Python, JavaScript, Go, FastAPI, SQL, MongoDB*

- Engineered **FastAPI and Go** backend services with MongoDB and SQL for a game serving **20,000+ concurrent players**, handling player-stat queries, command traffic, and leaderboard updates during peak activity
- Powered **100+ real-time leaderboards** and player-stat commands across a community platform serving **60,000+ users**, tracking rankings, player trends, tournaments, and server-wide activity

Educational Classification Benchmark | *PyTorch, DSPy, Scikit-learn, Embeddings*

- Built and benchmarked **logistic regression, neural network, embedding, and LLM** pipelines under a shared multiclass evaluation setup using identical training and test partitions for direct model comparison
- Achieved **80% test accuracy** with structured LLM function calling across **1,000+ labeled student responses**

TECHNICAL SKILLS

Languages: Python, JavaScript, TypeScript, C++, Go, Java, SQL

Frameworks & AI: React, Node.js, FastAPI, Flask, Flutter, PyTorch, Scikit-learn, DSPy, Hugging Face

Data & Tools: Supabase, PostGIS, MongoDB, Docker, Linux, AWS, CI/CD, Git, Selenium, BeautifulSoup