

PROFESSIONAL SUMMARY

Senior Computer Science student at Rensselaer Polytechnic Institute, graduating in Fall '26, focused on large language model (LLM) research and development, including tool- and agent-based frameworks (MCP & ACP style systems) and retrieval-augmented generation (RAG) pipelines. Seeking an entry-level AI research or applied LLM engineering role where I can contribute to model training, evaluation, and deployment.

EDUCATION

- **Rensselaer Polytechnic Institute** Troy, NY
Bachelor of Science in Computer Science Aug. 2024 – Dec. 2026
- **Suny Polytechnic Institute** Utica, NY
Bachelor of Science in Computer Science Aug. 2022 – April. 2024

RESEARCH & INDUSTRY EXPERIENCE

- **Griffiss Institute** Rome, NY
Research Intern – AI/LLM Research May 2025 – Aug 2025
 - **Large language model research:** Supported machine learning R&D focused on LLMs, designing and running experiments for model training, optimization, and information retrieval in defense and national-security-oriented applications. Reviewed recent LLM/RAG literature and analyzed results in terms of accuracy, latency, and robustness.
 - **Tool- and agent-based systems:** Designed and experimented with tool- and agent-based LLM workflows using MCP- and ACP-style orchestration and agent-to-agent coordination to enable multi-step pipelines and interaction with external tools and data sources, measuring task completion and robustness across scenarios.
 - **Retrieval-Augmented Generation (RAG):** Implemented retrieval-augmented generation (RAG) pipelines, including document ingestion, chunking, embedding, retrieval, and response generation; conducted ablation-style experiments on retrieval parameters and prompt strategies to evaluate trade-offs between retrieval quality, latency, and answer correctness on custom benchmarks.
 - **Model training with Unsloth:** Used the Unsloth framework to fine-tune LLMs on domain-specific datasets, iterating on training configurations and hyperparameters, performing structured experiment sweeps, and tracking metrics to improve model alignment, robustness, and downstream task performance.
 - **HPC and containerized workflows:** Packaged experiments in Docker to ensure reproducible environments and deployed them on a high-performance computing (HPC) cluster, managing jobs, resources, and experiment tracking for large-scale training and evaluation runs.
 - **Evaluation and documentation:** Developed scripts and evaluation harnesses, including LLM-as-a-judge evaluation, to compare model variants and RAG configurations; summarized experiment results and documented best practices for LLM/RAG development and deployment for future researchers and engineers.
- **LiftRyte Start-up** Troy, NY
Co-Founder & Front-End Engineer Jan 2024 – Present
 - **Product development:** Co-founded LiftRyte, a mobile app designed to help gym-goers improve their exercise form in real time by developing front-end components in React Native and shaping the overall user interface and experience.
 - **Team collaboration and current status:** Collaborated closely with a small team on feature design, implementation, and testing; the product is currently on hiatus, with active development paused while we evaluate next steps and long-term direction.

PROJECTS

- **JWT Authentication Sample Project (2024):** Developed a full-stack user authentication system using JSON Web Tokens (JWT) with registration, password hashing (bcrypt), and login functionality. Integrated MongoDB for persistent data storage and created API endpoints in Express.js for user registration, login, and protected routes, along with a responsive HTML/CSS/JavaScript frontend. *Technologies:* Node.js, Express.js, MongoDB, bcrypt.js, JWT, HTML, CSS, JavaScript.
- **EverQuest Item Finder Bot (2024):** Built a Discord bot that helps users find EverQuest items by combining natural language processing with structured database queries. Used the OpenAI API to translate natural language queries into SQL, designing prompt templates and parsing logic to robustly convert noisy user queries into structured SQL, retrieving item data from a PostgreSQL database, and exposed additional functionality through a lightweight Flask web interface. *Technologies:* Python, Flask, OpenAI API, Discord API, PostgreSQL, HTML/CSS.

SKILLS

- **Languages:** Python, JavaScript, Bash, Java, C, C++, HTML & CSS
ML/AI: vLLM, ollama, LM Studio, NumPy, HuggingFace Transformers, Unsloth
LLM Tooling: MCP, ACP, RAG, LLM-as-a-judge evaluation
DevOps/Data: Docker, Git/GitHub/GitLab, MongoDB, PostgreSQL