Yeji Lee

West Bloomfield, MI | leeyeji@msu.edu | 248.662.7245 | LinkedIn: yeji-lee2003 | GitHub: leeyeji2305
Resume (website): https://yejilee2305.github.io/

EDUCATION

Michigan State University — **East Lansing, Michigan**
Bachelor of Science in Computer Science, Minor in Business — *Expected Dec 2025*

- Dean's List
- Relevant coursework: Introduction to Programming I & II (python, C++), Computer Organization & Architecture, Algorithms & Data Structures, Software Design, Algorithm Engineering, Artificial Intelligence, Database Systems, Mobile App Development, Big Data Analysis

EXPERIENCE

Michigan State University — **East Lansing, Michigan**
Software Engineer — *August 2025 - Present*

- Built an end-to-end orchestration pipeline with FastAPI microservices connecting discovery, extraction, generation, and validation tasks, cutting manual integration time by 45%.
- Created a web crawler that scraped and normalized 35+ RFCs from a multitude of mirrors to automate network protocol discovery and the dataset creation process.
- Engineered multi-packet flow and TCP (Transmission Control Protocol) variant generators generating 3-5 packet exchanges per session and validated with Wireshark dissectors for accuracy.

Formula SAE, Michigan State University — **East Lansing, Michigan**
Mechanical / Software Design Member — *August 2022 - May 2023*

- Worked with the team to design and make a customized tire rack for the race car that increased overall functionality by 20% in the garage during the time of the pit.
- Collaborated with over 10 engineers on CAD modeling, part fabrication, and assembly.
- Helped implement simulation-based checks to validate the design optimization was meeting weight and stability testing requirements for the durability of the design when a load was applied.

Panera Bread — **East Lansing, Michigan**
Team Lead — *May 2024 – November 2025*

- Guided and trained a team consisting of 6-8 employees per shift while achieving 100+ transactions/hour with an order accuracy rate of 98%.
- Developed a task rotation system that decreased customer wait time by 2 minutes and improved workflow efficiency by 15%.
- Created a system that enforced safety and cleanliness standards, maintaining a perfect zero-incident record for all supervised shifts.

PROJECTS

TaskFlow - Full Stack Productivity App

- Developed a React + Node.js + FastAPI web application that allows users to create, tag, and track tasks across multiple projects with real-time updates.
- Developed RESTful APIs for CRUD operations, authentication, and analytics; integrated a PostgreSQL database using SQL Alchemy ORM.
- Deployed to AWS EC2 using Docker and NGINX reverse proxy, achieving 99.9% uptime and over 5,000 API requests per day during testing.

Packet Forge: AI Network Protocol Engine

- Created an Intent Based Packet Generating Application (PCAP) from User's requirements into Network (RFC compliant) Packet Flows, producing over 50 PCAPs for the purpose of testing, analysis and verification.
- Designed and built a Command Line Interface (CLI) application to allow developers to discover all RFCs defined for Protocols in a single source by integrating 3 sources of RFC mirror sites: normalizing over 80 Protocol Aliases and mapping over 35 Network Protocols to their respective RFC Documents, thus ensuring coverage of each RFC for the Protocol.
- Built a Multi-Packet Flow Generator that generates 5 to 12 Stateful Packets per Flow Session to simulate a real-life client to server communications and provides a means of generating fully repeatable test materials.
- Developed an application for generating multiple TCP traffic variants for each TCP Protocol, generating 3 to 4 different TCP Variants per Protocol (Fragmentation, Retransmission and Timing Jitter) and generating hundreds of TCP traffic variants to test TCP Protocols under various network conditions and to validate their performance and robustness.

Aquarium Simulation

- Created an interactive aquarium displaying inheritance and polymorphism in object-oriented design.
- Implemented 5+ fish behaviors (e.g., swim, chase, eat, flee), as well as add/remove actions at runtime, and supported 15+ dynamic objects interacting with an action in each simulation for realism.
- Optimized the rendering loop to improve frame rate by ~20% to reduce lag for demonstrations and conducted usability tests with classmates; 90% of users reported that their understanding of OOP principles improved after using the simulation.

SKILLS

Programming: Python, C++, Java, SQL, JSON, JavaScript, TypeScript
Frameworks / Tools: React, Node.js, FastAPI, Uvicorn, Docker, Kubernetes, GitHub, GitLab, VS Code, CLion, PuTTY, Postman
Cloud / DevOps: AWS, Linux, Bash scripting, CI/CD pipelines
Data / AI: TensorFlow, scikit-learn, Pandas, NumPy, Power BI, Excel
Networking / Security: Wireshark, PCAP/PCAPNG analysis, ABNF parsing, Network simulation, CompTIA Security+
Concept / Practices: API design/testing, OOP (Inheritance, Polymorphism), Agile, Scrum, Version Control (Git)