

Yash Priyadarshi

(646) 360-7222 | yashpriyadar@gmail.com | <https://www.linkedin.com/in/1402-yp/> | <https://github.com/yashp1402> | [Portfolio](#)

EDUCATION

The Pennsylvania State University

Master of Science in Computer Science Engineering | GPA: 3.83 / 4

State College, USA

Aug 2024 - May 2026 (Expected)

Vellore Institute of Technology

Bachelor of Technology in Computer Science and Engineering | GPA: 3.6 / 4

Vellore, IN

Jul 2018 - May 2022

WORK EXPERIENCE

Ericsson

Software Engineer II

Bengaluru, IN

Aug 2022 - Jul 2024

- Automated complex site deployment workflows for AT&T and CNIS using **Python** and **Golang**, reducing manual provisioning overhead by 80% and eliminating human-error configuration drifts.
- Led technical mentorship for incoming engineers on KSM architecture, **unblocking 5 critical release-blocking issues** and accelerating the overall product delivery **timeline by 20%**.
- Engineered a **Pre-Upgrade Resource Validator** for the CCD platform, **eliminating deployment failures** by enforcing strict resource allocation constraints prior to system upgrades.
- Architected a resilience testing framework in Golang to validate **Pod Disruption Budgets** across Bare Metal and CAPO clusters, ensuring **high availability** during node maintenance events.
- Hardened the CCD security posture by **remediating** 95% of **CVEs**, directly reducing security incident reports by 40%.

Software Engineer Intern

Jan 2022 - Jun 2022

- Developed a high-fidelity **RESTful API** simulator (OMC 2.0) using **Flask** and **Swagger**, decoupling frontend development from backend dependencies to accelerate parallel testing.
- Containerized the QA simulation environment using **Docker** and **Kubernetes**, reducing local environment setup time by 50% and ensuring parity between development and production stages.
- Orchestrated the deployment of the simulation suite using **Docker** and **Kubernetes**, establishing a portable runtime environment that reduced QA provisioning time by ~40%.

TECHNICAL SKILLS

- Languages:** Python, Go, C++, JavaScript/TypeScript, SQL, Java, Shell Scripting.
- Cloud & DevOps:** Kubernetes, Docker, Git, AWS EC2, Helm, Jenkins, Linux, CI/CD Pipelines, Distributed Systems.
- Backend & Web:** REST APIs, gRPC, Flask, FastAPI, Node.js, Next.js, Express.js, Tailwind, Prisma, Auth.js.
- Data & AI:** PostgreSQL, MongoDB, MySQL, GraphQL, TensorFlow, PyTorch, Sklearn, Numpy, Pandas, R, MATLAB.
- Coursework:** Deep Learning for NLP, Computer Vision, Operating Systems, Machine Learning, Internet of Things.

PROJECTS

Distributed Key-Value Store

Dec 2025

- Orchestrated a fault-tolerant, strongly **consistent** distributed storage engine in **Golang**, implementing the **Multi Paxos** consensus algorithm to guarantee **linearizability** across a 3-node cluster.
- Parallelized peer replication by constructing a non-blocking **gRPC** layer with **Go** channels, cutting write latency by 40% versus sequential processing.
- Fortified **data durability** against system crashes by integrating Write Ahead Logging and automated log replay, validating partition tolerance via **AWS EC2** stress testing.

SkillForge

Nov 2025

- Architected a **full-stack** application using **Next.js 16** and **PostgreSQL**, unifying reactive **React** components with server-side logic to eliminate N+1 fetching bottlenecks and slash page load latency by **40%**.
- Hardened API endpoints using **Zod** and **Auth.js**, reducing runtime type mismatches by 95% and blocking unauthorized access across all routes.
- Refactored UI architecture with **Tailwind CSS** and **shadcn/ui**, cutting client bundle size by 30% (via Next.js analyzer) and improving responsiveness across screen sizes.

NLP Research Agent

May 2025

- Synthesized a 3.2 K-example dataset of **LLM Prompt Optimization** Engineering abstracts by scraping arXiv and extracting problem-approach pairs via **few-shot prompting** with Llama-3.2-3B Instruct.
- Fine-tuned 1B-parameter models (**Llama-3.2**, **Gemma-3**) using **QLoRA** and **SFT** techniques, boosting **BLEU** scores by 15% while slashing inference latency by 8%.
- Quantified model performance across decoding strategies like **beam**, **nucleus**, **top-k** by benchmarking against standard **METEOR** and **ROUGE-L** metrics.