Tanay Shah

Raleigh, NC | +1 (984)-356-5527 | tshah27102002@gmail.com | linkedin.com/in/tanay-shah-a4a557213

Education

Master of Computer Science (GPA: 3.94/4) **Aug 2024 – May 2026**
North Carolina State University ***Raleigh, NC***
Coursework
Cloud Computing, Parallel Systems, Design of Analysis and algorithm, Computer networks, Software engineering

Bachelor of Technology in Computer Science and Engineering **Nov 2020 – June 2024**
Institute of Technology, Nirma University ***Ahmedabad, India***
Coursework
Data Structures and Algorithms, Database Management Systems, Computer Vision, Operating Systems, Object Oriented Programming, Machine Learning, Deep Learning, Data Mining, Big Data Analytics, iOS Development

Skills

Programming: Python, Java, JavaScript, TypeScript, R, C/C++, Kotlin, Swift, HTML
Web Technologies: React/Redux/Next.js, Node.js, Express.js, Flask, Django, FastAPI, GraphQL, CSS, HTML
DevOps and Virtualization: AWS, Azure, GCP, Terraform, Docker, Kubernetes, Jenkins, Apache Kafka, CI/CD Pipelines
Database: PostgreSQL, MongoDB, Oracle, Redis, SQL, NoSQL, Cassandra, DynamoDB
AI Libraries: TensorFlow, PyTorch, Keras, NumPy, Pandas, Scikit-learn, Matplotlib

Experience

Software Engineer Intern - *Nimble Data Technologies* **May 2025 – Aug 2025**
- Built **Model Context Protocol (MCP) servers** for SOC alert triage using **Python**, Chainlit, Pydantic, and **FastMCP 2.0**; integrated **Netskope** and **VirusTotal** via REST APIs; **automated enrichment** and severity scoring, cutting manual triage time by **60%** and processing **5,000** alerts per day.
- Designed **rules-based and LLM-assisted decision engine** with indicators of compromise (IOC) correlation and risk scoring; improved precision, **reduced false positives by 35%**, and lowered MTTR by **40%** across SIEM and SOAR workflows.
- Delivered a **production-ready platform** with HTTPS auth, secret management, structured JSON logging, OpenAPI/Swagger docs, and **GitLab CI/CD**; achieved **85% test coverage** with unit and integration tests, enabling weekly releases and stable deployments.

Back-end Developer Intern - *Crest Data System* **Jan 2024 – May 2024**
- Built a **Python automation tool** for API integration: generated project directories and pipelines; validated manifests with **JSON Schema**; verified endpoints; packaged secure ZIP archives with **API Key** and **Bearer Token** support—**reducing setup time by 40%** and improving developer throughput.
- Verified **API performance** across two distinct APIs and validation methods to ensure accurate data handling and integration; implemented **test-driven development** with **85% unit test coverage** and tracked work in **Jira** under Agile practices, improving reliability and observability.

Junior DevOps Engineer - *Capri Technosys* **Apr 2023 – Jan 2024**
- Managed and deployed **scalable cloud applications** on **AWS** with **Docker**, **Terraform** and **Kubernetes** maintaining **99.9% uptime**; automated **50+ cloud resources** in Terraform cutting deployment time and errors by **35%** as validated from **Jenkins** logs over three months.
- Built **Docker images** for 5 key applications and deployed on **Amazon ECS** to support **1,000+ daily users** with seamless scalability; implemented **CI/CD pipelines** with **Jenkins** and **GitHub Actions** reducing deployment time by **40%** and boosting reliability.

Projects

Cloud IDE | *Kubernetes, Docker, DynamoDB, React.js, Node.js, AWS, GCP, PostgreSQL* GitHub
- Engineered a **scalable cloud-based IDE** with Python and Node.js support, integrating an **AI-enhanced code editor** supporting up to **100 concurrent users**.
- Containerized and deployed **microservices** (Authentication, Resource Provisioning, UI) on **Google Cloud Run** for autoscaling; used **PostgreSQL** for projects/auth data.
- Used dedicated **Kubernetes pods** for secure, isolated code execution; **ingress controller** for dynamic routing; added **real-time code sharing** for multi-user collaboration.

DocXtract | *Colpali, RAG, OpenAI, React.js, Node.js* GitHub
- Built a **multi-document query-response system** using **GPT-4**, embedding pages as images to retain visual context and ensure accurate document interpretation.
- Leveraged **Colpali** for **page-level image extraction** and indexing to ensure precise document representation with visual elements preserved.
- Applied **retrieval-augmented generation (RAG)** to fetch relevant pages and integrate images into prompts, preserving graphical data for accurate interpretation.