

Vertika Singh

Noida, India | singhvertika119@gmail.com | +91 7523851576 | vertika-singh09

Professional Summary

Computer Science undergraduate specializing in AI/ML with hands-on experience in artificial intelligence, machine learning, computer vision, NLP, and backend development. Skilled in Python, TensorFlow, PyTorch, FastAPI, and Node.js with a strong foundation in data structures and problem-solving. Seeking AI/ML Development or Software Development internship opportunities

Education

JSS Academy of Technical Education, Noida, B.Tech in CSE (AI/ML)

Sept 2023 – Oct 2027

- CGPA: 8.1/10.0

Projects

Entropy AI- DevOps Log Analyzer (2026)

GitHub

- Built a multi-service log monitoring stack with Docker Compose (FastAPI API, log-ingestion agent, simulated app, Streamlit dashboard) that detects ERROR/CRITICAL events in real time and triggers automated incident analysis.
- Implemented a RAG-style RCA pipeline that matches errors to DevOps runbooks via keyword retrieval, then uses Groq (Llama 3.1) through LangChain to generate structured root-cause and remediation reports from rolling log context.
- Designed an event-driven ingestion flow with a rolling 20-line context window and REST API (POST /api/v1/analyze), plus a live dashboard for log streaming and manual RCA on pasted error snippets.
- **Tools Used:** Python, Groq API, LangChain, Docker, FastAPI

KOKO- Offline AI Agent (2026)

GitHub

- Deployed a local AI agent using Ollama (Llama 3B / Moondream 1.8B) and Streamlit, ensuring zero data leakage for private system tasks
- Built a custom agentic loop with deterministic routing and regex safety fallbacks to execute Python (psutil) and SQL commands accurately
- Bypassed native JSON formatting limits of small models by engineering an XML-tagging pipeline to securely query local SQLite databases.
- **Tools Used:** Python, Pydantic, Streamlit, Ollama, Llama 3.2, FastAPI, Moondream, SQLite

UI -> Code Generator (2026)

GitHub

- Built a full-stack “UI → code” tool that accepts UI screenshots via drag-and-drop, calls a vision LLM (Gemini) from a FastAPI backend, and returns React + Tailwind component code.
- Implemented a REST API (POST /api/generate) with multipart image upload, CORS for local dev, and a markdown/JSX parser so the client receives clean, runnable component strings.
- Developed a React (Vite) + Tailwind frontend with Sandpack for live in-browser preview, resizable panels, and full-screen preview so generated components can be inspected immediately.
- **Tools Used:** Python, React, Vite, Tailwind, FastAPI

Technical Skills

Languages: Python(primary), Java, C++

CS Fundamentals: Data Structures and Algorithms, OOP, DBMS

ML / AI: TensorFlow, PyTorch, Keras, scikit-learn, OpenCV, Computer Vision, Transfer Learning, NLP, RAG, AI Agents, LlamaIndex, Prompt Engineering

GenAI and LLMs: Ollama, LlamaIndex, RAG pipelines, n8n Automation

Backend: JavaScript, FastAPI, Node.js, Express.js, REST APIs, JWT Authentication, Webhooks, CRUD Operations

Frontend: HTML, CSS, React, Streamlit(python)

Databases: MongoDB, SQL

Cloud & DevOps: Docker, Render, Git, GitHub, Google Colab