

Swastik Agnihotri [Portfolio Website](#) | [LinkedIn Profile](#)

Phone: 9026986801 | Email: swastikwork007@outlook.com



Professional Summary

DevOps Engineer | InfoZ IT Services | *July 2024 – Present*

- Inherited fragile deployment pipelines that broke on every other release. Rebuilt them from scratch with staged rollouts and automated rollback triggers — deploy failures dropped to near-zero within the first month.
- Audited infrastructure spend and found 40% of running instances were oversized or idle. Right-sized compute, introduced scheduled scaling, and cut monthly cloud costs without touching availability.
- Replaced a dozen manual runbooks with scripted automation. What used to take an engineer's Friday afternoon now runs unattended in under 3 minutes.
- Stood up structured monitoring with alerting that actually means something — tuned thresholds, silenced noise, made on-call survivable. Mean time to detection went from "user reported" to under 5 minutes.
- Eliminated environment drift between dev, staging, and production by containerizing services and enforcing config parity. "Works on my machine" stopped being an acceptable excuse.

Technical Lead | LDM College | *October 2023 – July 2024*

- Designed and shipped a full college ERP (library, exams, attendance, finance) that handles 500+ daily active users — and still does, months after handoff. Not a demo. Not a prototype.
- Architected for bursty load patterns: near-idle most of the day, then hundreds of concurrent requests during attendance windows and exam submissions. No over-provisioning, no crashes.
- Reworked the backend to process exam submissions under sustained peak load without queuing failures. Identified the bottleneck was synchronous DB writes, moved to batched async inserts — submission time dropped from 8s to under 1s.
- Rewrote the heaviest database queries (attendance aggregation, fee lookups) that were doing full table scans. Added composite indexes and materialized views where it mattered — page loads went from painfully slow to instant.
- Built automated scraping pipelines that pull external academic data directly into the system on schedule. Removed a manual process that used to eat 6+ hours per week of staff time.
- Owned the full delivery — architecture decisions, code reviews, deployment, and production stability. When something broke at 11pm during exam week, my phone rang.

Site Reliability Engineer | HCL Technologies | *March 2023 – September 2023*

- Took over a system with a weekly average of 3+ SEV-2 incidents. Focused on eliminating repeat failures rather than faster response — brought recurring incidents down by identifying and fixing the actual root causes, not just restarting services.
- Automated the operational toil that was consuming 30% of the team's capacity: log rotation, certificate renewals, health check restarts. Freed up engineering hours for work that actually matters.
- Overhauled alerting rules that had devolved into pure noise — 200+ alerts/day, most ignored. Cut it to <20 actionable alerts. On-call engineers started actually reading them.
- Improved system performance by profiling and tuning what was already running — connection pool sizing, garbage collection tuning, query plan analysis — instead of defaulting to "add more instances."

Frontend Developer | LDM College | *June 2022 – March 2023*

- Built the initial frontend for the college management system — dashboards, data views, and workflows — used by faculty and admin staff daily with zero formal training needed.
- Designed data dashboards that replaced weekly manual reports. Staff went from "I think attendance is low" to seeing real numbers in real time.
- Wrote web scrapers to automate external data collection that was previously done by hand in spreadsheets. Reduced a 2-day manual process to a 10-minute automated job.
- Profiled and fixed frontend performance issues — lazy-loaded heavy components, debounced API calls, trimmed bundle size. Load times cut in half on the low-spec machines the college actually uses.

Research Intern | IIT Kanpur | *May 2019 – July 2019*

- Built a test automation utility that reduced manual verification effort in a research data pipeline — small tool, but it replaced a step that was eating 2 hours per test cycle.

- Handled data preprocessing and transformation for an active research project — learned that production-grade code isn't about cleverness, it's about reliability and clarity.

Education

B.Tech in Computer Science

Rajkiya Engineering College, Bijnor, Uttar Pradesh

July 2018 – June 2022

Skills

- **DevOps & Cloud:** CI/CD Pipelines (Rollbacks, Staged Rollouts), Docker, AWS (Resource Optimization, Cost Management), Monitoring, Alerting, Observability, Infrastructure Tuning, Linux.
- **Programming & Scripting:** Python (Pandas, Flask, OpenPyXL, Regex), SQL, SAP ABAP.
- **Automation & Web Scraping:** Selenium, Scrapy, BeautifulSoup, Test Automation.
- **Databases:** PostgreSQL (Optimization, Indexing, Materialized Views).
- **Backend & Architecture:** API Integration, ERP Design, Performance Tuning, Code Review.
- **Data & Visualization:** Power BI, Tableau, Matplotlib, Data Preprocessing.
- **Version Control:** Git, GitHub.

Certifications

- Post Graduate Certification in AWS + DevOps, IIT Guwahati July 2025 – Jan 2026
- Google IT Automation with Python | [Certificate](#)
- Complete Python Bootcamp | [Certificate](#)

Projects

Enterprise-Level Web Scraper for E-commerce Analytics

- Started with scattered, manually collected competitor data that was inconsistent and quickly outdated.
- Designed a scraping pipeline combining Scrapy for scale and Selenium for dynamic content, instead of relying on a single approach.
- Handled data for 10,000+ products, structuring it into usable datasets using Pandas rather than raw dumps.
- Built the system to run reliably over time, dealing with site changes and failures instead of breaking silently.
- Result: Teams could actually track pricing and availability trends continuously, instead of reacting late or working with incomplete data.

Corporate Sales Reporting Dashboard

- Inherited a reporting process that depended heavily on manual Excel work and was prone to delays and inconsistencies.
- Replaced it with a Python-driven pipeline using Pandas and OpenPyXL, focusing on accuracy and repeatability rather than just automation.
- Designed reports to highlight trends and patterns, not just raw numbers, so they could be used for decisions.
- Automated delivery via email, removing dependency on manual sharing.
- Result: Reporting became consistent, faster, and actually useful for decision-making, not just documentation.

Advanced Email Marketing Automation System

- Started with generic, one-size-fits-all campaigns that had low engagement.
- Built a system that triggers emails based on user behavior and actions, instead of fixed schedules.
- Integrated SendGrid API for scalability and reliability instead of managing raw SMTP flows.
- Focused on making the system adaptable, so campaigns could evolve without rewriting logic.
- Result: Campaigns became more targeted and meaningful, improving engagement without increasing manual effort.

Robust API Integration Framework

- Faced repeated effort every time a new external API needed to be integrated.
- Designed a reusable Flask-based integration layer, instead of writing custom code for each API.
- Standardized handling of JSON/XML parsing, error handling, and retries, reducing integration friction.
- Built it with failure scenarios in mind, not just success cases.

- Result: New integrations became faster and more predictable, without introducing instability.

Automated Document Management System

- Dealt with unstructured file storage where documents were hard to locate and manage.
- Built a rule-based system using regex and file handling libraries to categorize and organize files automatically.
- Designed it to handle edge cases (naming conflicts, duplicates) instead of failing silently.
- Result: Reduced manual effort and brought structure and consistency to document handling, making retrieval easier.

Social Media Management Suite

- Started with fragmented posting and no clear way to track performance across platforms.
- Built a centralized system using Selenium and APIs to schedule and manage posts instead of manual uploads.
- Added basic analytics to understand what content performs, not just publish blindly.
- Focused on reliability, ensuring posts go out even if platforms behave inconsistently.
- Result: Teams could plan, execute, and evaluate content in one place, improving consistency and engagement.