**Kushagar Garg**
Email: kushagar340@gmail.com
Github | LinkedIn | LeetCode

## Professional Summary

B.Tech CS (AI/ML) student at Newton School of Technology. Active open source contributor building fixes and benchmarks in Python, JavaScript, and Bash. Experience spans CLI tooling, infrastructure-graph sync, CI/CD automation, and scientific-computing benchmarking. Comfortable across Linux, Docker, and REST API workflows.

## Education

- **Newton School of Technology** — Sonipat, Delhi-NCR
  *B.Tech in Computer Science (AI/ML); CGPA: 8.125* — *Aug. 2025 – May 2029*

## Experience

- **Open Source Contributor**
  *MDAnalysis, AWS CLI, Google, Anthropic, CNCF, SakanaAI* — *Aug 2025 - Present*
  - MDAnalysis (Python): Added support for reading unit cell dimensions from CRYSIN records to the MOL2 parser and built a performance benchmark for HydrogenBondAnalysis. Built a benchmark for the PDB reader.
  - AWS CLI (Python): Resolved a NameError crash in shorthand-parsing error messages, shipped in release v1.45.34. Corrected `configure get` to properly print section values instead of erroring.
  - Claude Code, Anthropic (Bash): Diagnosed and resolved a silent failure in the issue-triage automation pipeline, where a script exited under `set -euo pipefail` without printing a diagnostic message.
  - Cartography, CNCF (Python): Resolved a Route53 sync bug where DNS-to-IP relationships were never materialized in the infrastructure graph. Migrated the AWS OpenSearch intel module off a deprecated boto3 client.
  - Google Migration Planner (Python): Traced and resolved a TypeError that was masking real Microsoft Graph API error messages during batch-request logging.
  - SakanaAI & sktime (Python): Corrected an averaging bug in a reinforcement-learning reward computation. Shipped three maintenance fixes to sktime's skbase library covering a broken pre-commit dependency, test warnings, and sparse-matrix comparison logic.

## Projects

- **filedrop**: Published npm CLI (`@dreamstick/filedrop`, v2.0.8 – 6 stars, 8 forks) for instant file transfer over a local network via QR code. Files are encrypted client-side with AES-256-GCM before ever reaching the network; the decryption key travels only in the QR code's URL fragment, so it never touches the server. Adds IP rate limiting and path-traversal protection.
- **Blockchain-v1**: From-scratch blockchain node built in Python and Flask – block structure, proof-of-work mining from a mempool, wallets, and transactions. Verified working peer-to-peer across multiple physical machines on the same network.
- **sculpt-cli**: CLI for image-to-3D and text-to-3D generation, auto-routing requests across six open models hosted on free Hugging Face Spaces, with circuit breakers, rate limiting, and content-addressed caching. Shelved once Spaces backend uptime made it impractical to rely on.

## Programming Skills

**Languages:** Python, Typescript, JavaScript, Node.js, Rust, Solidity

**Backend:** Flask, REST APIs, CLI design, structured error handling

**AI/ML:** LLM fine-tuning (QLoRA, Unsloth), local inference (Ollama), Hugging Face, `gradio_client`

**Infra & DevOps:** Docker, Kubernetes, GitHub Actions CI/CD, Linux, Oracle Cloud, AWS

**Tools:** Git, GitHub, VS Code, Freebuff CLI