# – AI Skills test

## AssignmentTitle

Data Cleaning and Validation for Network Inventory Records

## Background

’s core products manage DNS, DHCP, and IP Address Management (DDI) data at scale. Clean, consistent network data is critical for automation and insight. In this challenge, you’ll simulate building an intelligent data‑cleaning assistant that validates and normalizes a small, messy dataset representing network inventory. You’ll apply both deterministic rules and AI reasoning to demonstrate your problem‑solving approach.

## Objective

Transform the provided **inventory\_raw.csv** into a cleaned and normalized **inventory\_clean.csv** suitable for downstream IPAM/DNS workflows. You’ll validate, normalize, enrich, and flag anomalies using your own rules and optionally an LLM of your choice. You are allowed to use an LLM to generate the code as well if you wish to do so. Watch out for bugs in the generated code. If you are unclear on any of the requirements, feel free to clarify using an LLM.

## Files Provided

- inventory\_raw.csv – Synthetic, intentionally messy dataset.  
- run\_ipv4\_validation.py.txt – Example IPv4 validation logic.  
- TEMPLATES/ – Folder with empty templates to fill:  
 - prompts.md – record of prompts and iterations  
 - approach.md – your pipeline description  
 - cons.md – known limitations or risks  
 - run.py.txt – simple orchestrator to extend with your logic  
- README.md – schema and reference info

## Tasks

1. Validation

* For each record, validate and normalize key fields: IP address, MAC address, hostname, FQDN, owner, device type, and site. Add flags like ip\_valid, hostname\_valid, fqdn\_consistent, and device\_type\_confidence where relevant.

1. Normalization

* Apply deterministic transformations—lowercasing, trimming, deduplication, and canonicalization. Derive fields like subnet\_cidr and reverse\_ptr for valid IPs.

1. Anomaly Reporting

* Generate anomalies.json listing row ID, affected field(s), issue type, and recommended action.

1. AI Involvement

* Use an LLM to resolve ambiguous cases (e.g., unclear device\_type or owner). Record every prompt in prompts.md along with rationale. Keep temperature low (≤ 0.2) and outputs structured (JSON preferred).

1. Limitations & Reflection

* In cons.md, describe at least three limitations or trade‑offs (e.g., model hallucination risk, missing external context, split‑horizon FQDN unmodeled).

1. Reproducibility

* Provide a single entry point (run.py or run.sh) that regenerates inventory\_clean.csv and anomalies.json.

## Expected Output Schema

|  |  |
| --- | --- |
| Column | Description |
| ip, ip\_valid, ip\_version, subnet\_cidr | IP fields and validity |
| hostname, hostname\_valid, fqdn, fqdn\_consistent, reverse\_ptr | Naming validation |
| mac, mac\_valid | MAC normalization |
| owner, owner\_email, owner\_team | Ownership parsing |
| device\_type, device\_type\_confidence | Classification |
| site, site\_normalized | Site normalization |
| source\_row\_id, normalization\_steps | Traceability |

## Submission Package

Zip and submit the following files:

* inventory\_clean.csv
* anomalies.json
* prompts.md
* approach.md
* cons.md
* run.py or run.sh

## Alternatively, you may create a Github repo and share the link with us.

## Final Notes

We’re not scoring for perfection—focus on clear reasoning, reproducibility, and awareness of trade‑offs. Show us how you balance deterministic logic with responsible use of AI.