🧪 Case Study: Company Name Standardization Across Multiple Systems

📌 Background

You are working as a Data Engineer in a mid-sized enterprise that integrates data from multiple internal and external systems. A recurring issue is inconsistent formatting of **company names** across:

1. **System A** – A CRM platform (API-based)
2. **System B** – A financial system (API-based)
3. **Internal Excel Sheet** – Used by the sales/admin teams to manually track changes

Each source refers to the same companies but with different naming conventions, such as:

System A	System B	Excel Reference
ACME Pte Ltd	ACME Limited	Acme Inc.
Alpha Tech Holdings	Alpha Technologies	ALPHA TECH
Global-X Corporation	Global X Corp.	Global X

Inconsistencies in naming cause issues in reporting, data integration, and downstream analytics. Additionally, any change in one source is not automatically reflected in others, resulting in duplicated records and incorrect data merges.

🎯 Objective

Your task is to design a solution to **detect, clean, and standardize company names across all sources**, and propose a scalable method to **maintain consistency** when data is updated from any of the three systems.

🔧 Your Deliverables

The case study submission should include the following:

1. Approach & Assumptions (Documentation)

- Describe your approach to:
  - Identifying matches and mismatches
  - Resolving discrepancies in company names
  - Creating a **“master reference table”** of canonical names
- Detail your assumptions (e.g., language casing, abbreviations, etc.)
- Discuss how you would handle new or changed company names going forward
- Describe how changes will be synced across sources

2. Technical Implementation (Code + Sample Files)

- Provide **Python code** that:
    - Loads sample data from all 3 sources (you may simulate API responses and Excel input)
    - Maps inconsistent names to a canonical format using logic or mapping
    - Outputs a **cleaned dataset** that can be used for analytics
- Optional: Include logic for fuzzy matching or manual override mapping

3. Data Flow Diagram or Architecture (Visual)

- A high-level diagram showing how data flows between sources and where standardization occurs (ETL process, staging, master data table, etc.)

4. Scalability Proposal

- A brief plan for:
    - Handling new company names over time
    - Maintaining this mapping table (manual vs. automated)
    - Integrating this process into a CI/CD pipeline or data platform (optional but appreciated)

📁 Suggested Format

Please submit your case study as:

- A ZIP or Git repo with:
    - README.md
    - standardize_names.py (or Jupyter notebook)
    - Sample input files (excel.xlsx, system_a.json, system_b.json)
    - Output file(s)
- A PDF/Word document with your explanation and visuals (optional diagrams)

⏱️ Timeline

You will have **1 week** to complete and submit this case study before your interview.