**Design test planning and execution of a software product that checks the validity of any other product**

**Validity check to be done on the parameter - Expiry Date**

## **G-1 Group**

**Date:- 25th July 2025**

|  |  |
| --- | --- |
| **Group Participants** | Points |
| **Jyotirmay Chowdhury(coordinator) - 30639** | **1-2** |
| **Nandhu -30400** | **3-4** |
| **Abhigyan Maji- 30382** | **5-6** |
| **Shikha - 31136** | **7-8** |
| **Gourab Das - 30392** | **9-10** |

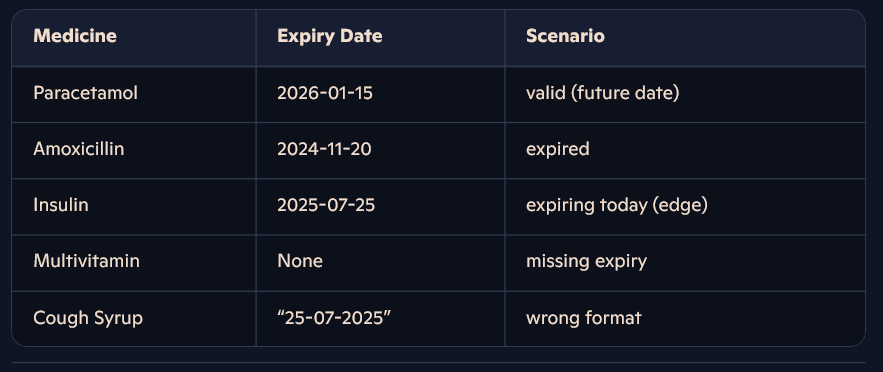
**1. Define Objective**

- Identify medicines whose expiry date is before today

- Fail products if expiry\_date < current\_date

- Pass products if expiry\_date >= current\_date

**2. Collect Sample Test Data**

****

- Paracetamol – Expiry: 2026-01-15 (valid future date)

- Amoxicillin – Expiry: 2024-11-20 (already expired)

- Insulin – Expiry: 2025-07-25 (expiring today)

- Multivitamin – Expiry: None (missing date)

- Cough Syrup – Expiry: “25-07-2025” (invalid format)

**3. Choose Test Environment**

- Language: Python (for date parsing and comparison)

- Framework: Pytest (optional, for structured test cases)

- Tools: Date libraries, test runner, logging utilities

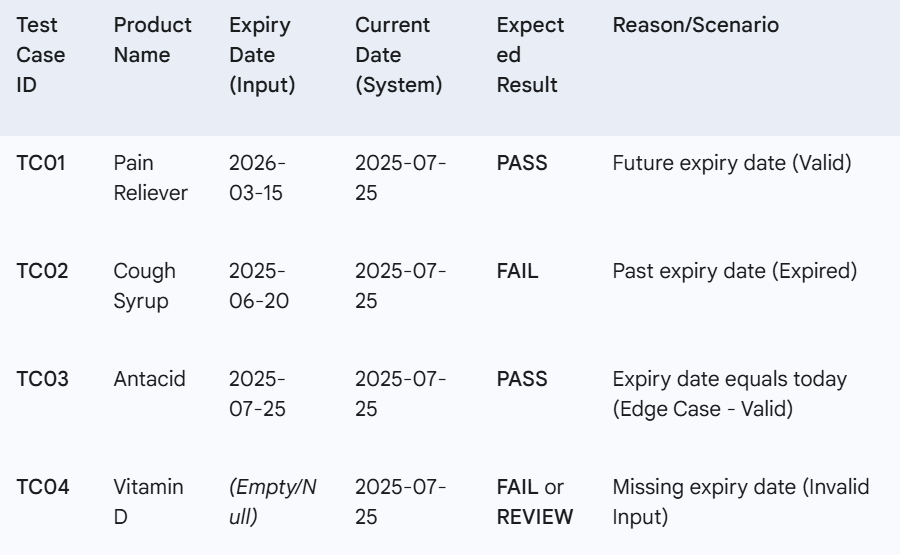
**4. Build Expiry Validation Logic**

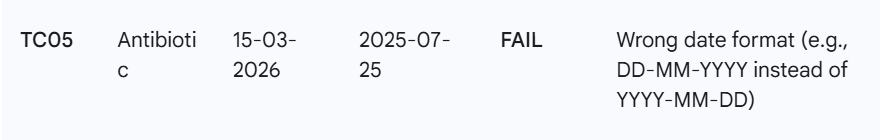
- Parse expiry date string using a consistent format (YYYY-MM-DD)

- Compare parsed date to the current system date

- Return a boolean or status flag (valid vs. invalid)

- Handle parsing errors and null values gracefully





**5. Design Test Cases**

- TC01: Future expiry date → Expected “PASS”

- TC02: Past expiry date → Expected “FAIL”

- TC03: Expiry date equals today → Expected “PASS” (edge case)

- TC04: Missing expiry date → Expected “FAIL” or “REVIEW”

- TC05: Wrong date format → Expected “FAIL” (format error)

**6. Plan Test Execution**

- Iterate through each medicine record

- Apply expiry validation logic

- Log or record the result as PASS/FAIL

- Include timestamp, product name, expiry value, and status

**7. Extend with Automation**

- Parameterize test cases in Pytest or another test framework

- Use fixtures for sample medicine lists

- Generate automated test reports after each run

**8. Reporting & Alerts**

- Summarize results in CSV, JSON, or database table

- Create Power BI or Excel dashboards to visualize pass/fail trends

- Trigger email or messaging alerts for failed/near-expiry items

**9. Edge Case Handling**

- Null expiry dates → auto-fail and flag for manual review

- Non-parseable strings → log as “format error” and fail

- Future data anomalies (e.g., year far ahead) → review data consistency

**10. Review & Optimize**

- Validate that all realistic medicine scenarios are covered

- Scale logic for bulk data ingestion (CSV/XLSX files)

- Profile performance on large catalogs and optimize date comparisons

- Periodically audit test data and update scenarios as new edge cases emerge

# Thank You