Case Study Solution – Backend Engineering Intern

Candidate: Vishwajeet Kadam

PART 1 – CODE REVIEW & DEBUGGING

Issues Identified:

1. No input validation
2. SKU uniqueness missing
3. Two commits lead to inconsistency
4. No rollback
5. Incorrect price handling
6. No HTTP status codes
7. Duplicate inventory entries possible

Corrected Code Reasoning:
- Added validation
- Added SKU check
- Single DB transaction
- Rollback on failure
- Proper responses

(Full code in create_product_fixed.py)

PART 2 – DATABASE DESIGN

Tables:
companies(id,name)
warehouses(id,company_id,name)
products(id,sku,name,price,type)
suppliers(id,name,email)
product_suppliers(product_id,supplier_id)
inventory(product_id,warehouse_id,quantity)
inventory_history(id,product_id,warehouse_id,change,timestamp)
bundle_items(parent_product_id,child_product_id,quantity)

Questions to clarify:
- Threshold per product?
- Should bundles auto-update?
- Should inventory history track reason?

PART 3 – LOW STOCK API

Endpoint: GET /api/companies/{company_id}/alerts/low-stock

Logic:
1. Retrieve warehouses
2. Join products + inventory
3. Filter quantity < threshold
4. Add supplier info

5. Compute days_until_stockout

Sample Response:

```
{
'alerts':[{'product_id':123,'current_stock':5}],
'total_alerts':1
}
```