

## Part -3

Assumptions (about schema and logic)

Database Schema (simplified):

Companies

- id

Warehouses

- id

- name

- company\_id

Products

- id

- name

- sku

- product\_type\_id

ProductTypes

- id

- name

- low\_stock\_threshold

WarehouseInventory

- id

- warehouse\_id

- product\_id

- current\_stock

## Sales

- id
- product\_id
- warehouse\_id
- sale\_date

## Suppliers

- id
- name
- contact\_email

## ProductSuppliers

- product\_id
- supplier\_id

## Business Logic Assumptions:

- “Recent sales activity” = sales in the last 30 days.
- days\_until\_stockout is estimated based on average daily sales over the last 30 days.
- Multiple suppliers are allowed but only the primary one is returned.



## ✅ Implementation (Python + Flask)

```
from flask import Flask, jsonify
```

```
from datetime import datetime, timedelta
```

```

app = Flask(__name__)

# Mock DB calls (Replace with actual DB ORM/queries)
def get_low_stock_alerts(company_id):
    # Placeholder: should query actual DB
    today = datetime.today()
    recent_days = today - timedelta(days=30)

    # Sample data returned from DB for explanation
    # Each product-warehouse combo with recent sales
    data = [
        {
            "product_id": 123,
            "product_name": "Widget A",
            "sku": "WID-001",
            "warehouse_id": 456,
            "warehouse_name": "Main Warehouse",
            "product_type": "Widget",
            "low_stock_threshold": 20,
            "current_stock": 5,
            "total_sales_last_30_days": 12,
            "supplier_id": 789,
            "supplier_name": "Supplier Corp",
            "supplier_email": "orders@supplier.com"
        }
        # ... more rows
    ]

    alerts = []

```

```
for row in data:
```

```
    avg_daily_sales = row["total_sales_last_30_days"] / 30
```

```
    days_until_stockout = (
```

```
        int(row["current_stock"] / avg_daily_sales) if avg_daily_sales > 0 else None
```

```
    )
```

```
if row["current_stock"] < row["low_stock_threshold"]:
```

```
    alerts.append({
```

```
        "product_id": row["product_id"],
```

```
        "product_name": row["product_name"],
```

```
        "sku": row["sku"],
```

```
        "warehouse_id": row["warehouse_id"],
```

```
        "warehouse_name": row["warehouse_name"],
```

```
        "current_stock": row["current_stock"],
```

```
        "threshold": row["low_stock_threshold"],
```

```
        "days_until_stockout": days_until_stockout,
```

```
        "supplier": {
```

```
            "id": row["supplier_id"],
```

```
            "name": row["supplier_name"],
```

```
            "contact_email": row["supplier_email"]
```

```
        }
```

```
    })
```

```
return {
```

```
    "alerts": alerts,
```

```
    "total_alerts": len(alerts)
```

```
}
```

```
@app.route('/api/companies/<int:company_id>/alerts/low-stock', methods=['GET'])
```

```
def low_stock_alerts(company_id):
    try:
        response = get_low_stock_alerts(company_id)
        return jsonify(response)
    except Exception as e:
        return jsonify({"error": str(e)}), 500

if __name__ == '__main__':
    app.run(debug=True)
```

#### Edge Cases to Consider

1. No recent sales: Products with no sales in last 30 days are excluded.
2. No supplier information: If supplier data is missing, either return null or skip.
3. Zero average sales (division by zero): Prevent division by zero in days\_until\_stockout.
4. Data integrity: Ensure product belongs to the company's warehouses.
5. Multiple warehouses per company: Must check inventory by warehouse, not just product level.

#### Summary

- Endpoint: GET /api/companies/{company\_id}/alerts/low-stock
- Pulls data from products with recent sales in company warehouses
- Compares stock vs product-type threshold
- Returns supplier info and days until estimated stockout

```
127.0.0.1:5000/api/companies/1/alerts/low-stock - Brave
127.0.0.1:5000/api/companies/1/alerts/low-stock
Pretty-print
{
  "alerts": [
    {
      "current_stock": 10,
      "days_until_stockout": 150,
      "product_id": 125,
      "product_name": "Food Item C",
      "sku": "FUD-003",
      "supplier": {
        "contact_email": "sales@freshfoods.com",
        "id": 790,
        "name": "Fresh Foods Inc"
      },
      "threshold": 15,
      "warehouse_id": 456,
      "warehouse_name": "Main Warehouse"
    },
    {
      "current_stock": 15,
      "days_until_stockout": 90,
      "product_id": 124,
      "product_name": "Gadget B",
      "sku": "GAD-002",
      "supplier": {
        "contact_email": "orders@supplier.com",
        "id": 789,
        "name": "Supplier Corp"
      },
      "threshold": 20,
      "warehouse_id": 456,
      "warehouse_name": "Main Warehouse"
    },
    {
      "current_stock": 5,
      "days_until_stockout": 25,
      "product_id": 123,
      "product_name": "Widget A",
      "sku": "WID-001"
    }
  ]
}
```