RMMS Spring Boot

To Do: Need PGCRON/pgagent for auto deduction

* **Real-time Inventory Monitoring:**
  + Continuous monitoring of inventory levels.->can reverse engineer on raw material(stock) and material update
  + Automated alerts for low stock thresholds to ensure timely restocking.-> can be tracked with rawmaterials(stock vs threshold)
  + Significant reduction in stock-outs and menu unavailability.->handled by auto recommending stock reductions
* **Dynamic Menu Item Management:**
  + Manages menu item availability for both dine-in and online platforms. Can abstract and treat as one group entity-> that is the restaurant
  + Flags or temporarily disables dependent menu items when a critical raw ingredient level is reached, until replenishment occurs.-> remove menu item from menu when stock<threshold
  + Prevents customer dissatisfaction due and ensures smoother operational flow.
* **Automated Restocking & Supplier Management:**
  + Restocking schedules planned based on historical consumption data and predictive analytics.
  + Automated notification or suggestion of vendors/suppliers based on configured preferences and lead times.
  + Enables a proactive rather than reactive inventory strategy.
* **Detailed Raw Material Tracking:**
  + Categorization of all raw materials.-> schema available
  + Unit-wise tracking implemented (e.g., kilograms, liters, packets).-> schema available
  + Wastage tracking and generation of variance reports to identify discrepancies and improve purchasing decisions.->**NEED TO ADD A WASTAGE TABLE**
* **Operational Benefits:**
  + Enhances overall kitchen efficiency.
  + Minimizes human error in inventory management.
  + Achieves better cost control.

What I Should do:

Database:

**SCHEMA:**

Restaurant: (ID,Name,Location)

RawMaterials: (MaterialID ,RESTID,Name, Unit, Threshold, ExpiryPeriod, CurrrentStock, AutoRestock(bool),RestockPeriod ,reorderquantity)

RestID-> from restaurant id and restock period will alert restaurant every certain amount of time if not refreshed unit can be a enum

RestaurantSupplier(RestaurantId,supplierid) MANY TO MANY RELATION

Suppliers: (SupplierID,name,contactinfo)

materialOrder (materialId, supplierid, placedate, recievedate, ORDERID, deliveryPeriod, orderAmount, receivedAmount,status) ->

deliveryperiod of each material can be used to recommenf ideal supplier materialid is from rawmaterials supplierid from suppliers

MaterialUpdates ( shipmentID, supplierid,materialID,units,dateadded,status)-> supplierid from suplliers materialid from materials, status is enum with used,wated,unused

the wastage can be calculated with currentstock from rawmaterials subtracting units from it till less than 0 all the shipments that come after <0 show wastage and can be updated accordingly, check can be run everyday.

If supplierid is null then we can enforce stock<0 implying that it is actually a usage and not restock

Wastage(wasteid,expirydate,stockwasted,materialid,shipmentid)

as calculated from materialupdates,will be updated here, all wasted records in materialupdated are marked as used immediately

Good candidates for indexing:

* RawMaterials(RESTID) (since you’ll filter by restaurant often)
* MaterialOrder(MaterialID, SupplierID)
* MaterialUpdates(MaterialID)
* Wastage(MaterialID)

**Restaurant**:  
(ID, Name, Location)

**RawMaterials**:  
(MaterialID, RESTID, Name, Unit, Threshold, ExpiryPeriod, CurrentStock, AutoRestock, RestockPeriod, ReorderQuantity)

* RESTID → Restaurant(ID)
* Unit can be ENUM (kg, liters, pcs, etc.)-> here it is a schema CREATE TABLE units(unit\_name varchar(50) PRIMARY KEY);

**RestaurantSupplier**:, used as contacts with schema (restaurantid,supplierid,phone\_supplier)  
(RestaurantID, SupplierID)

* Many-to-many relation.
* Both are FKs: RestaurantID → Restaurant(ID), SupplierID → Suppliers(SupplierID).

**Suppliers**:  
(SupplierID, Name, ContactInfo)

* No restID here (handled by RestaurantSupplier).

**MaterialOrder**:  
(ORDERID, MaterialID, SupplierID, PlaceDate, ReceiveDate, DeliveryPeriod, OrderAmount, ReceivedAmount, Status)

* ORDERID = PK
* MaterialID → RawMaterials(MaterialID)
* SupplierID → Suppliers(SupplierID)

**MaterialUpdates**:  
(ShipmentID, SupplierID, MaterialID, Units, DateAdded, WasUsed, WasWasted)

* ShipmentID = PK
* SupplierID → Suppliers(SupplierID) (nullable → usage entry)
* MaterialID → RawMaterials(MaterialID)

**Wastage**:  
(WasteID, ExpiryDate, StockWasted, MaterialID, ShipmentID)

* MaterialID → RawMaterials(MaterialID)
* ShipmentID → MaterialUpdates(ShipmentID)