Recipe Suggestion System Logic

Overview

The recommendation engine connects your inventory with the **Sugran** recipe database. It uses a sophisticated matching algorithm that handles: - Ingredient Normalization (handling "tomatoes" vs "tomato") - Fuzzy Matching (handling "chopped onions" vs "onion") - Smart Availability Sorting (prioritizing recipes you can cook *now*)

Logic Flow

```
sequenceDiagram
    participant User as User's Device
    participant API as /api/recipes/suggestions
    participant DB as Supabase Inventory
    participant Sugran as Sugran Service

    User->>API: Request Suggestions

    par Fetch & Filter Data
        API->>DB: Get Active Inventory
        DB-->>API: User's Ingredients
        API->>API: Filter Expired Items
        API->>Sugran: Fetch Recipes (Limit 100)
        Sugran-->>API: Recipe List
    end

    loop For Each Recipe
        API->>API: Normalize Ingredients

        rect rgb(240, 248, 255)
            note right of API: Matching Logic
            API->>API: Check Exact Match
            API->>API: Check Fuzzy Match (>60% ratio)
            API->>API: Identify Missing Staples
        end

        API->>API: Calculate Availability Score
        note right of API: Score = Matched / Total Ingredients
    end

    API->>API: Sort by Availability Score (Desc)
    API->>API: Select Top Matches
    API-->>User: Return Suggested Recipes
```

Detailed Matching Process

1. Data Fetching

- **Inventory**: Fetches your non-consumed, non-expired items from Supabase.
- **Recipes**: Fetches a batch of vetted recipes from the Sugran API (`https://sugran.vercel.app/api/recipes`).

2. Ingredient Normalization

Before comparing, ingredients are "normalized" to ensure accurate matching: - **Canonical Names**: Converts variations like "diced tomatoes" -> "tomato". - **Inventory Mapping**: Creates a standardized list of what you own.

3. The Matching Algorithm

For every ingredient in a recipe, the system performs two checks against your inventory:

1. **Exact Match**:
   - Recipe: "Cumin Seeds" | You have: "Cumin Seeds" [x]
2. **Fuzzy / Substring Match**:
   - Recipe: "Fresh Cilantro Leaves" | You have: "Cilantro"
   - **Rule**: The match must share a substring AND have a length ratio > 0.6 (60%) to prevent false positives (like "corn flour" matching "flour" incorrectly).

4. Smart Scoring & Sorting

Recipes are ranked based on an **Availability Ratio**:

$$\text{Score} = \frac{\text{Ingredients You Have}}{\text{Total Ingredients in Recipe}}$$

- **Staple Handling**: Common items like *Salt, Oil, Water, Sugar, Pepper* are tracked.
- **"Exact Now" Tag**: If you are missing ONLY staples (max 1), the recipe is marked as "Cookable Now".

Code Reference

The core logic resides in `src/app/api/recipes/suggestions/route.ts`.