INNOVATEX Hackathon Project Brief – Part 1 (Pre-Hack Requirements)

SDG 2 & SDG 12: AI-Powered Food Management & Sustainability Platform

Context

This document contains the pre-hack (Part 1) requirements for the hackathon project. Part 2 (onsite challenges and AI requirements) will be revealed on the hackathon day.

Theme & Problem

Many individuals and communities face challenges with food waste, inefficient consumption, and limited access to nutritious meals on a budget. Your solution should support SDG 2: Zero Hunger, by improving food security and nutrition, and SDG 12: Responsible Consumption and Production, by reducing waste and promoting sustainable practices through mindful food tracking and planning.

High-Level Concept

Build a full-stack web application where users (individuals, families, or community groups) can create profiles, log food usage, manage inventories, and access basic tools for tracking consumption. In Part 1, focus on a solid foundation: authentication, profiles, data models, image upload for future scanning, and simple (non-AI) logging and reporting.

General Technical Guidelines

You can pick any technology or tech stacks.

- Full-stack web app using any modern stack.
- Persistent database for users, food logs, inventories, and resources.
- Clean, responsive, and usable UI.
- Deployed or easily runnable locally with clear instructions.

PART 1 – REQUIRED FEATURES (TO BE COMPLETED BEFORE ONSITE)

1. Authentication & User Management

- Implement user registration and login (email/password or similar secure method).
- Basic validation (e.g., required fields, invalid email format, password length).
- Store at minimum:
  - Full name
  - Email
  - Household size or dietary preferences (e.g., vegetarian, budget range)
  - Location (for future local features)

2. User Profile & Consumption Logging

- Create a Profile page for each user.
- Users must be able to:
  - Add or edit basic details like budget preferences and dietary needs.
  - Log daily food usage manually (e.g., item name, quantity, category like vegetable/dairy).
  - Manage a simple inventory list (add/remove items with quantities and dates).
- Additionally:
  - Provide a simple way to note or store consumption history (no AI/processing required in Part 1; just store this data).

3. Food Items & Inventory Database (Seeded Data)

- Create and seed a food items/inventory collection or table.
- Each item entry should include at least:
  - Item name
  - Category (e.g., fruit, dairy, grain)

        - Typical expiration period (e.g., days)
        - Sample cost per unit
    - Include a minimum of 15–20 relevant entries, focused on common household foods.
    - Build a simple Inventory page with:
        - List view of items
        - Basic filter options (e.g., by category, expiration)
        - Item details view.

4. Resources for Sustainable Practices (Seeded Data)

- Create and seed a resources collection or table for tips on waste reduction and nutrition.
- Each resource should include:
    - Title
    - Description or URL
    - Related category (e.g., waste reduction, budget tips)
    - Type (e.g., article, video)
- Include at least 15–20 resources mapped to common themes (e.g., meal planning, storage tips).
- Show these resources on a dedicated page or section.

5. Basic Tracking Logic (Non-AI)

- Implement a simple rule-based tracking feature:
    - Use the user's logged consumption and inventory.
    - Show basic summaries (e.g., total items in inventory, recent logs).
    - Recommend resources based on simple matches (e.g., if dairy logged, suggest dairy storage tips).

        - Show why a resource is recommended (e.g., “Related to: Dairy category”).
        - This logic can be simple but must be consistent and transparent.

6. Image Upload for Food Scanning (UI Only)

- Implement an upload interface for receipts or food labels (support JPG/PNG).
- Store uploads in database or storage (no processing required in Part 1).
- Allow users to associate uploads with inventory or logs manually.

7. User Dashboard & UI

- Create a dashboard/home view for logged-in users that includes:
    - Quick view of profile, recent logs, and inventory.
    - Basic summaries (from your tracking logic).
    - Recommended resources.
- Ensure navigation is clear (e.g., Navbar with: Dashboard, Logs, Inventory, Resources, Profile, Logout).
- The design should prioritize clarity, accessibility, and real usability over heavy visuals.

8. Documentation & Code Quality

- Include a README in your repository with:
    - Project overview (2–3 lines).
    - Tech stack used.
    - Setup steps (how to install dependencies and run frontend/backend).
    - Any environment configuration notes.
    - Seed data usage instructions.

- Code should be organized logically (separate routes/controllers/models where applicable).

What Teams Must Bring to the Onsite Hackathon

1. A working full-stack application implementing all Part 1 requirements.
2. Code pushed to a public or shareable repository.
3. Ability to run the project on a new machine using the README.

Important Notes

1. No advanced AI features are required in Part 1.
2. However, design your data structures and flows so that AI and smarter logic can be integrated in Part 2 without rewriting everything.
3. Teams that come with an incomplete Part 1 base will be at a disadvantage during the onsite hackathon.