PantryPal: Complete Application Functionalities

1. FUNCTIONALITY 1: AI-POWERED INVENTORY INGESTION

Goal: Make adding groceries effortless and magical. **User Goal:** "I just came back from the store. I want to add all my groceries to the app in less than a minute."

1.1. AI Enhancement

Use Optical Character Recognition (OCR) on grocery receipts and image recognition for fresh produce.

1.2. Full Pipeline

1.2.1. User Interface (UI)

- The user taps a large "+" button on the home screen.
- Presented with three clear options:
  - **Scan Receipt** (AI-powered "wow" feature)
  - **Scan Barcode** (quick single-item entry)
  - **Add Manually** (reliable fallback)

1.2.2. Scan Receipt Pipeline

Flutter Frontend:

- Opens the phone's camera.
- User takes a clear, well-lit photo of the grocery receipt.
- The image is compressed and sent to the backend.

Backend (AI Processing):

1. **OCR Service:** The image is processed through an OCR model (e.g., Google Cloud Vision or a fine-tuned open-source model) to extract all text.
2. **Text Parsing & Structuring:** A custom AI script or LLM (e.g., GPT-4) parses the text, identifies items, quantities, and prices, and removes irrelevant data (store name, tax, etc.).
3. **Product Matching:** Parsed items are matched against a food database to standardize product names, categories, and default shelf-life estimates.

Return to Frontend:

- Backend returns a structured list of identified items.
- UI displays a confirmation screen: "We found these 15 items. Please review."
- User corrects any mistakes, then taps "Confirm" to add all items to their digital pantry.

2. FUNCTIONALITY 2: DYNAMIC RECIPE & MEAL PLAN GENERATION

Goal: Provide personalized meal ideas based on available ingredients. **User Goal:** “It’s 6 PM. I’m tired and uninspired. Show me a simple, healthy recipe I can make right now with the ingredients I have, especially using the chicken that expires tomorrow.”

2.1. AI Enhancement

Use a fine-tuned Large Language Model (LLM) to generate unique and personalized recipes beyond database matching.

2.2. Full Pipeline

2.2.1. User Interface (UI)

- Button: **Generate a Recipe**
- Optional filters: “Quick (under 30 mins),” “Healthy,” “Vegan,” or a natural language input like “something for the kids.”

2.2.2. Backend (AI Processing)

Step A: Prepare the Prompt

```
"You are a world-class chef creating a recipe for a home cook.
Ingredients available: [Chicken breast (expires in 1 day), 3 tomatoes, 1 onion
   , half a lemon, olive oil, salt, pepper, rice].
User preferences: [Healthy, Quick (under 30 mins)].
Generate a unique, easy-to-follow recipe including a creative name,
step-by-step instructions, and cooking times. Prioritize using the chicken
   breast."
```

Step B: LLM Generation Prompt is sent to a powerful LLM (e.g., GPT-4 or Gemini).

Step C: Structure the Output Response is formatted as JSON:

```
{
  "recipeName": "Lemon Chicken Rice Bowl",
  "ingredientsList": [...],
  "instructions": [...],
  "cookTime": "25 mins"
}
```

Return to Frontend:

- Display recipe in a clean layout.
- User can tap *I made this!*, triggering automatic ingredient deduction from inventory.

3. FUNCTIONALITY 3: THE “PANTRYPAL” AI ASSISTANT (CHATBOT)

Goal: Enable full conversational control of the app. **User Goal:** “I don’t want to tap through menus. I just want to ask questions and get answers.”

3.1. AI Enhancement

A conversational chatbot capable of understanding intent and executing functions such as inventory checking, recipe generation, or adding to the shopping list.

3.2. Full Pipeline

3.2.1. User Interface (UI)

- A chat icon is always visible on-screen.
- Tapping it opens a familiar conversational interface.

3.2.2. Backend (Intent Recognition)

- User query is analyzed via intent recognition or LLM function-calling.
- Examples:
  - "Do I have any eggs left?" → `check_inventory`
  - "Add bread to my shopping list." → `add_to_shopping_list`
  - "I feel like Italian tonight." → `generate_recipe`

3.2.3. Action & Response Generation

- **check_inventory:** Returns item quantity and expiration.
- **add_to_shopping_list:** Updates list and confirms addition.
- **generate_recipe:** Triggers Functionality 2 pipeline with user context.

3.2.4. Return to Frontend

- Text response displayed in chat.
- Larger results (e.g., full recipe) shown as rich embedded cards.

4. FUNCTIONALITY 4: PROACTIVE & ACTIONABLE EXPIRATION ALERTS

Goal: Turn expiring food notifications into valuable, actionable insights. **User Goal:** "Remind me before food expires and show me an easy way to use it."

4.1. AI Enhancement

Generate contextual alerts that not only notify but also suggest recipes to use expiring items.

4.2. Full Pipeline

4.2.1. Step 1: The Background Task (The Watcher)

- A scheduled task runs daily (e.g., 8 AM).
- Scans inventory for items expiring within 1–3 days.
- Produces a list of expiring items with user IDs.

4.2.2. Step 2: AI-Powered Suggestion Generation (The Brain)

Prompt Example:

```
"User 'Khalifa-Bouneb' has avocados that expire in 2 days. He also has onions,
    tomatoes, and limes.
Generate:
1. A short push notification (<150 chars) encouraging usage.
2. A simple recipe title.
Tone: friendly and helpful."
```

AI Response (JSON Example):

```
{
  "notification_message": "Your avocados are perfectly ripe!  Don't let them
     go to waste. How about some fresh guacamole tonight?",
  "suggested_recipe_title": "5-Minute Classic Guacamole",
  "target_item": "avocado"
}
```

Intelligent Bundling: When multiple items are expiring, the AI combines them for joint recipe ideas (e.g., "Your chicken and bell peppers are expiring soon! Perfect for Chicken Fajitas ").

4.2.3. Step 3: Sending the Push Notification

- Backend sends notification via Firebase or Apple Push Service.
- Hidden metadata includes the suggested recipe title.

4.2.4. Step 4: The In-App Experience

- User taps the notification → app opens.
- Metadata triggers Functionality 2 using the suggested recipe.
- The full recipe (e.g., "5-Minute Classic Guacamole") appears instantly.

Summary: This pipeline transforms a basic alert into an intelligent loop from *problem* (expiring food) to *solution* (ready-to-make recipe) in one seamless user interaction.