

# Menu Generation Assistant

Flutter + Supabase App

---

Haotian Sun, Dominic Pörtl, Joshua Lympany

**University of Tübingen**

23.05.2025

Department of Marketing

# Outline

Applications

Front End

Back End

# Applications

---

# User Story Examples (1)

## **Image Upload**

Upload a photo of my ingredients → get a recipe.

## **Diet Preferences**

App! I am vegan, or I am looking to lose fat.

## User Story Examples (2)

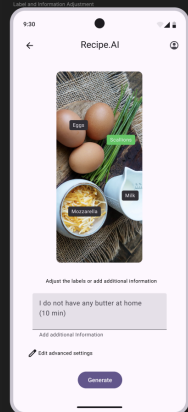
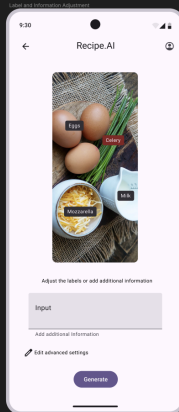
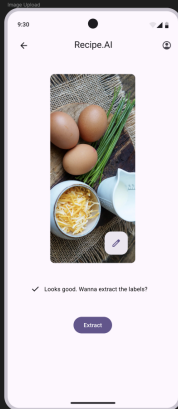
### **Ingredient Edit**

Adjust detected ingredients (even AI can make mistakes)

### **Recipe Share**

Share my generated recipe via link with friends.

# Basic Idea



# Potential Applications (1)

## **Smart Cooking Guide**

Personalized step-by-step instructions based on ingredients

## **Fast Cooking**

Instant recipe suggestions for meals under 15 minutes

## Potential Applications (2)

### **Meal Planning & Tracking**

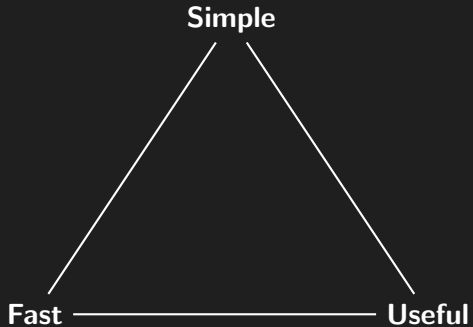
Auto-generate weekly menus from photos

### **Minimalist Recipes**

Create tasty dishes using only 3–5 ingredients on hand



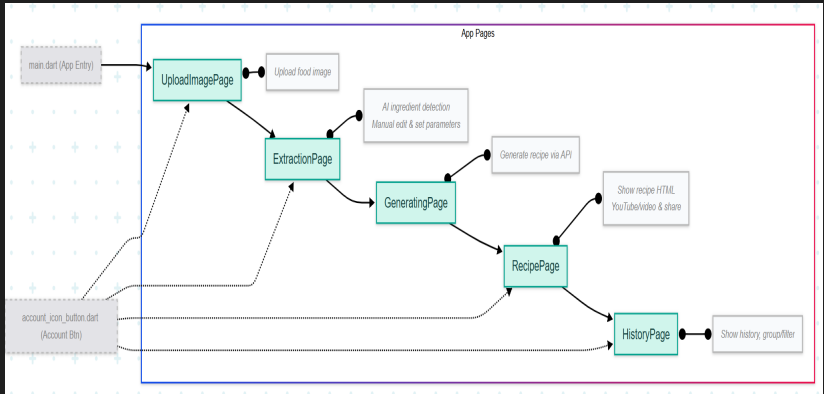
# Value Proposition



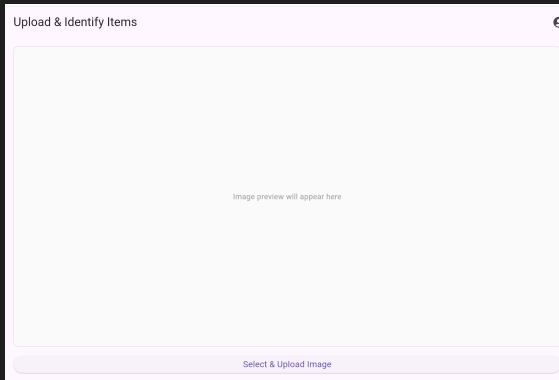
**Front End**

---

# App Structure

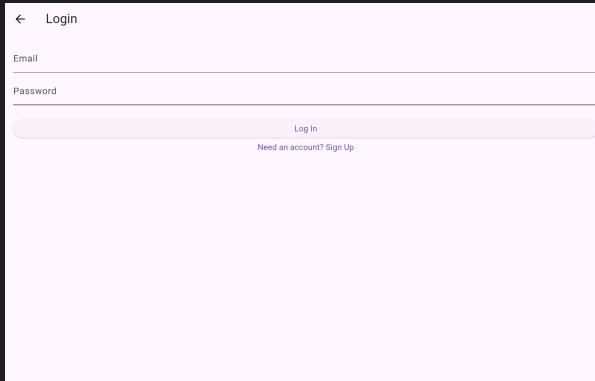


# Initial Page



- Unlogged-in user selects and uploads a food image to start the process.

# Log-in Page



← Login

Email

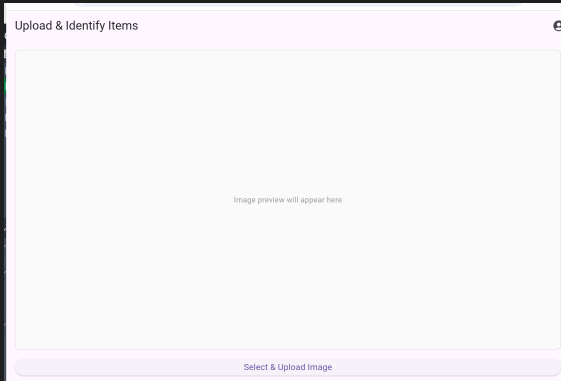
Password

Log In

[Need an account? Sign Up](#)


- Users can securely log in or sign up with their email and password to access personal features like history.


# Upload Page



- Logged users select and upload a food image to start the recipe generation process.

# Extraction Page

← Adjust Items & Options 



**Recipe Options:**

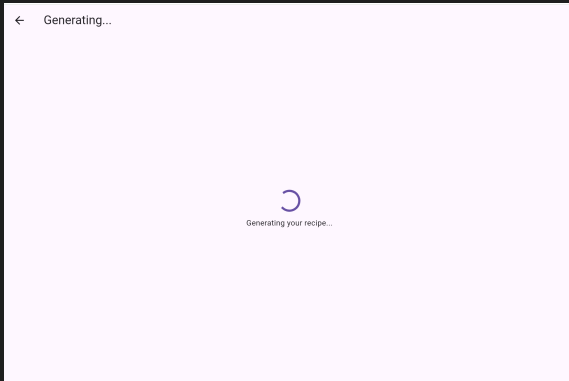
Meal Type:  Dietary Goal:

Meal Time:  Amount of People:

[Generate Recipe](#)

- Users review and adjust AI-detected ingredients, and set preferences for the recipe.

# Generating Page




- The app calls the edge function from backend to generate a recipe based on the selected ingredients and user preferences.



# Recipe Page

[←](#) Quick & Healthy Breakfast Bowl [👤](#)



[Watch Cooking Video](#)

2. Wash and slice a quarter of the cucumber into small pieces.

3. Wash and slice 1-2 mini peppers into rings.

4. Wash 10 grapes and halve them.

5. Portion one yogurt into a bowl.

6. Arrange the orange slices, cucumber pieces, pepper rings, and halved grapes on top of the yogurt.

7. Enjoy immediately! You can eat the baby carrots on the side as a snack.

Estimated Calories: Approximately 300-350. (Varies depending on yogurt and orange size.)

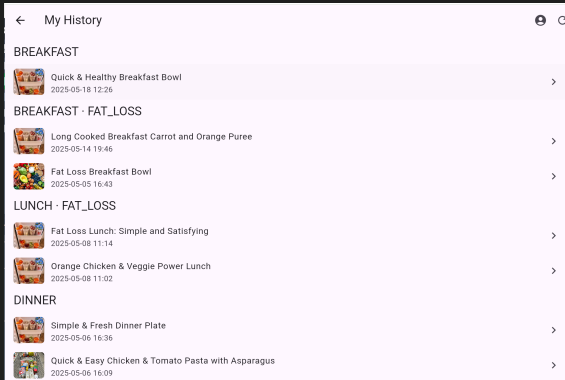
This breakfast bowl is a quick and refreshing way to start your day with a mix of fruits, vegetables, and protein.

[Share Recipe Text](#)

[Refine & Re-generate](#)

- Shows the generated recipe with steps, nutrition info, a video link, and sharing options.

# History Page



- Displays your recipe history, grouped by meal type and dietary tags for easy access.

# History Table

history	
user_id	uuid
image_url	text
meal_type	text
dietary_goal	text
detected_items	jsonb
recipe_html	text
id	uuid
created_at	timestampz
video_url	text
amount_people	text
meal_time	text
restrict_diet	text
tags	_text
recipe_title	text

- All recipe generations are stored in a structured table, linked to user accounts, with key fields for filtering and display.

# Back End

---

## First Question: How do we develop the Backend?

Consideration	Custom Backend	Out-of-the-Box Solution
Customizability	High	Moderate to High
Time Investment	Significant (Dev + Ops)	Low (Setup + Config)
Opportunity Cost	Higher (focus on infrastructure)	Lower (focus on core product)
Pricing Model	Variable (infra + dev)	Often Tiered (Free to Enterprise)
Open-Source Nature	Own Code	Varies (some are, some aren't)

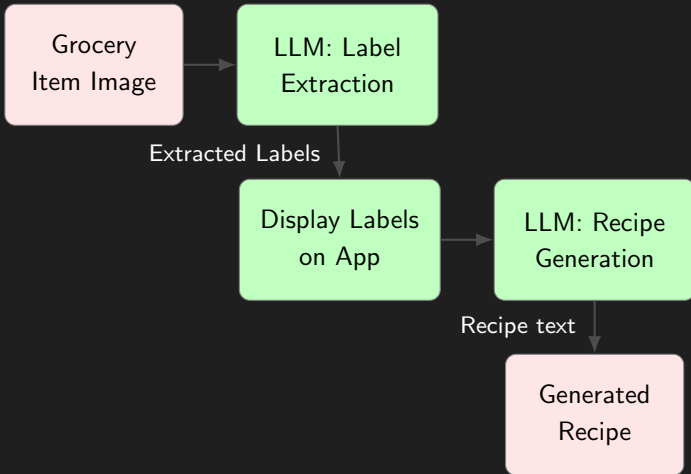
# How Supabase addresses these questions

## Benefits of using Supabase

1. Rich out-of-the-box functionality
2. Low time-investment
3. Focus on core product features instead of infrastructure  
(Automated hosting by Supabase)
4. Generous free tier
5. Open-source code

## Second Question: How do we extract the labels and generate the recipe?

### Process Overview: From Image to Recipe



# How can we extract the labels consistently?

## The magic happens through Prompt Engineering:

- **Precision in Detection:**

*"Only include items that are clearly identifiable as food. Ignore any non-food items or objects whose edibility is ambiguous."*

- **Focus on the Item, Not Packaging:**

*"Don't include information about the packaging... We are interested in the item itself, not the container."*

- **Structured and Reliable Output:**

*"Return results in valid JSON, no extra text. Ensure 'quantity' is an integer."*



# The End

Thank you for listening!