Produktions-Prompt für GhibliKitchen (.jsx Generierung)

Anweisung:

Kein Begleittext, keine Erklärung. Wenn dein Interface Codefences nutzt, dann **genau eine** Codefence mit ```jsx und erster Zeile:

```
// Datei: Woche-{{WEEK_NR}}-{{START_DATE}}.jsx
```

Ziel & UI-Layout (fix)

- Tabs: "Kochbuch" (A4 quer) und "Einkaufsliste" (A4 hoch).
- TopBar-Buttons (nur drei): "PDF erzeugen", "HTML exportieren", "Drucken". Kein "HTML öffnen".
- Nach PDF-Erzeugung: Download-Link unter dem jeweiligen Tab-Inhalt anzeigen.
- Jede Rezeptseite ist eine Seite (1 Rezept = 1 Seite), Panel links $\leq 1/3$, Rezept rechts $\geq 2/3$.
- DALL·E-Prompts **NICHT** rendern (nur im Code als Strings verfügbar).
- Bilder-Uploads (Cover + je Rezept) via FileReader, persistiert in localStorage.
- Pro Rezept oberhalb des Titels die Überschrift "{Wochentag} {Morgen|Mittag|Abend}" anzeigen.
- Mittags kein Metformin-Reminder; Frühstück/Abend mit Reminder.
- Bei jedem Rezept eine Kurzbeschreibung (Ursprung/Art des Gerichts + "inspiriert von ...").

Fixe Metadaten & Dateibasis

```
export const meta = { title: "Woche {{WEEK_NR}}", startDate:
  "{{START_DATE}}", id: "woche-{{WEEK_NR}}-{{START_DATE}}" }
const FILE_BASE = "Woche {{WEEK_NR}} {{START_DATE}}";
// UI-Titel:
// "GhibliKitchen - Woche {{WEEK_NR}}"
// "GhibliKitchen - Einkaufsliste - Woche {{WEEK_NR}}"
```

Farb- & Stilvorgaben

Nur HEX/RGBA, keine oklab/oklch/lab/color-mix.

```
const COLORS = {
  pageBg:"#FAF7F1", text:"#111827", border:"rgba(0,0,0,.10)",
  panelBG70:"rgba(255,255,255,.70)", panelBG80:"rgba(255,255,255,.80)",
  white:"#FFFFFF", emerald:"#059669", amber:"#f59e0b",
  sky:"#0284c7", neutral:"#404040", indigo:"#4f46e5", btnShadow:"0 6px 20px
rgba(0,0,0,.12)"
};
```

- Card-Styles: cardPanelStyle (links) und cardMainStyle (rechts) mit borderRadius:18 , Padding 20/22, boxShadow: COLORS.btnShadow, zusätzlichem Außenabstand.
- Grid: 12 Spalten; Panel span 4; Rezept span 8.
- Klassen/CSS: _.page (jede Druckseite, page-break-after: always), _.avoid-break (break-inside: avoid), _.print:hidden (Buttons/Uploads verbergen).

PDF/HTML-Export

```
    Lade html2pdf.js@0.10.1 via CDN (ensureScript).
    html2canvas Pass 1: { scale:3, useCORS:true, background:'#FAF7F1', letterRendering:true, foreignObjectRendering:false }.
    Fallback (Blob < 50 KB): foreignObjectRendering:true, letterRendering:false; pagebreak:{ mode:['css'], after:['.page'] }.</li>
    jsPDF: { unit:'pt', format:'a4', orientation:'landscape' } für Kochbuch; 'portrait' für Einkaufsliste.
    pagebreak (Kochbuch): { mode:['css','legacy'], after:['.page'], avoid: ['.avoid-break'] }.
    HTML-Export: eigenständige .html mit eingebettetem CSS (gleiche Farben/Seitenregeln) erstellen und als Download anbieten.
```

Gesundheits- & Küchenregeln

- Küchenmix: CN/JP/KR dominant (mind. 6/7 Tage), max. 1 leichtes IT-Gericht.
- Pro Rezept (2 Personen): 60-90 g KH gesamt; Protein-Hinweis optional (20-40 g p. P.).
- Garmethoden: **Dämpfen, Sieden, Schmoren**; wenig Öl; Zwiebel/Knoblauch sparsam & gut gegart; **Säure mild**; Algen/Jod **sparsam**.
- Schwangerschaft: **kein Rohes**; Fisch/Ei vollständig **durchgaren**; quecksilberarme Fische; Sojasauce **natriumarm**.
- Titel: Deutsch + Originalname + Schriftzeichen.

Datenstruktur

- Exakt 7 Tage × 3 Meals = 21 Rezepte, IDs: mo|di|mi|do|fr|sa|so + -f|-m|-a|.
- Objekt je Rezept:

```
id, // z. B. "mo-f"
title, // Deutsch + Originalname + Schriftzeichen
desc, // Kurzbeschreibung mit Ursprung + "inspiriert von …"
target, // KH gesamt (2 P.) + optional Protein p. P.
ingredients: [ ≥5 Einträge mit Mengen (g/ml/TL/EL/Stück/l) ],
steps: [ ≥3 schonende Schritte ],
checks, // "Gastritis √/ ↑ … · Diabetes √/ ↑ - ≈XX g KH · Schwangerschaft
√/ ↑ …"
swaps, // sinnvolle Alternativen
side, // milde Beilage/Drink
remind // Frühstück/Abend true, Mittag false
}
```

• PROMPT_HEADER String-Konstante (nicht rendern) für Bildprompts; je Rezept prompt: buildPrompt(PROMPT_HEADER, "...").

Einkaufsliste

- LIST_SUMMARY gruppiert:
- "Protein/Fisch/Tofu"
- · "Gemüse/Pilze"
- "Reis/Nudeln/Sättigung"
- "Algen/Brühen/Würze"
- Mengen **arithmetisch** aus den Zutaten aufsummieren, Einheiten **vereinheitlichen** (ml statt l, "Reis (roh)"/"Reis (gekocht)" etc.).

Bild-Handling

• Uploads für **Cover + je Rezept** via FileReader; persistiere DataURLs in localStorage (Key: \$ {FILE_BASE}::img::...).

Tests (Pflicht, am Ende mounten)

```
    7 Tage, 21 Meals, eindeutige IDs.
    mealLabel Mapping korrekt; Zusatzfunktionen: dayLabel(id) → {Montag..Sonntag}, mealTitle(id) → {Morgen/Mittag/Abend}.
    Metformin-Logik: Frühstück/Abend true, Mittag false.
    Jedes Rezept ≥5 Zutaten + ≥3 Steps.
    Keine verbotenen CSS-Funktionen (oklab/oklch/lab/color-mix) in Strings.
    FILE_BASE Regex: ^Woche {{WEEK_NR}} \d{4}-\d{2}-\d{2}$.
    buildPrompt("A", "B") === "A\nB".
    LIST_SUMMARY besitzt genau 4 Gruppen.
    Konsolen-Log: "[GhibliKitchen] All tests passed (JSX)."
```

Ergebnis

Nur die **.jsx** ausgeben.