

ChefsCart — Comprehensive Product Spec & Design System (v0.4)

Mission Ship the fastest, most delightful path from “*What should I cook?*” to a ready-to-checkout Instacart cart — powered by an AI sous-chef that truly learns the user week-to-week.

0 Project Snapshot

Item	Detail
Brand / Domain	ChefsCart (chefscart.ai)
Elevator pitch	“Your AI sous-chef that turns personal meal plans into a ready-to-checkout Instacart cart in ≤ 5 min.”
Launch KPIs (Beta)	30 % visitors reach cart-review · 20 % click Instacart checkout · NPS ≥ 60 by week 4
MVP scope lock	Instacart IDP public endpoints only; no ChefsCart fees in free-beta

1 Feature Narrative (share with Claude Code)

1. Silently verify ZIP coverage. Unsupported → wait-list capture.
 2. Conversational wizard (or free-form chat) gathers: scope, meal types, diet rules, culinary style, pantry & inspiration photos.
 3. Call GPT-4o-mini with function-calling to generate **N + 40 %** meals (for fallback). Return recipe cards with hook, macro, cook-time, cost.
 4. User may swap/remove; on approve, store plan → build Instacart list via [/products/products_link](#) (stub if no keys).
 5. If >10 % items unmatched, regenerate or prompt user.
 6. Deep-link returned via chat & Resend email; regen requests handled conversationally.
-

2 Assistant Goals & Required Data

#	Objective	Data captured	Why it matters
1	Ensure serviceability	ZIP (geo-IP autofill)	Abort early if no coverage

2	Define workload	Meals per week, people per meal	Drives recipe & portion count
3	Guardrails	Diet type(s), allergies, avoid list, organic/budget	Filters generator & store search
4	Context	Meal types, max cook time, difficulty, cuisines, preferred stores	Shapes complexity & sourcing
5	On-hand inventory (opt)	Pantry photo(s) vision parse	Prevent duplicate buys, lower cost
6	Creative cues (opt)	Inspiration image(s) / keywords	Personalises menu tone
7	Generate plan	N + backups, macros, cost	Provides fallback slack
8	Finalise	User confirms / swaps	Locks weekly plan
9	Build list	Aggregate ingredients → Instacart	Produces shoppable cart
10	Deliver	Deep-link + email	Smooth hand-off & support

3 Brand Design System

Token	Value	Usage
Primary Orange 600	#E65300	CTA buttons, progress bars
Primary Orange 50	#FFF1E6	Selected states bg
Accent Lime 500	#A3D34E	Success badges
Accent Sky 500	#4BB8F2	Links, info chips
Neutral 100	#FFFAF2	App background
Neutral 900	#2F3438	Headlines
Border-radius	12px (buttons), 24px (cards)	Friendly look

Shadow-1 0 2px 4px Cards, modals
 rgba(0,0,0,.06)

Spacing Scale 4 / 8 / 12 / 16 / 24 / 32 / 48 / 64 Vertical rhythm

Typography Inter family — SemiBold for H1-3, Regular 16/24 px body, tracking -0.02 em.
Optional display accent : Caveat.

Icons Lucide 1.5 px stroke in brand palette.

4 UI Improvement Back-log & Bug List

Landing / ZIP-Gate

1. Container max-w 480 px, auto-center.
2. Input/label vertical gap → `gap-y-4`.
3. Instacart availability message inline inside input (`inline-flex icon+text`).
4. Fix placeholder contrast (#6B6F72 → AA).

Chat Wizard (Step 1 of 9)

1. Step label mis-aligned below 320 px; ensure `flex items-center gap-2`.
2. Meal-type grid equal widths `min-w-[120px]` & `gap-3`.
3. Selected state: orange 50 bg + orange 600 border; focus ring 2 px sky.
4. Continue button full-width, disabled style & spinner.

Global

- 16 px side-padding ≤ md.
- Dark-mode toggle.
- Persist wizard state in `localStorage`.

- ARIA live-regions for validation errors.

Bugs

ID	Component	Symptom	Severity
B-01	Wizard step	Progress resets on reload	High
B-02	Button grid	Dessert button wraps on iPhone SE	Med
B-03	CTA state	Continue disabled after selection intermittently	High

5 Detailed User-Flow Matrix

#	Scenario	Entry Conditions	Key Screens	Decision Points	Success Path	Error / Fallback
1	New user, ZIP OK	No cookie, Instacart available	Landing → Wizard 1-9 → Review → Create cart	Confirm plan?	Redirect to Instacart link	API timeout → show skeleton + retry; hard-fail → save draft & email link
2	New user, ZIP not supported	No cookie, Instacart unavailable	Landing → ZipError modal	Enter new ZIP?	Loop to Landing	Wait-list form (Resend email)
3	Returning user, same week	<code>localStorage.profile.completed & weekHash === current</code>	Dashboard → Quick Review	Edit plan?	Same as 1	n/a
4	Returning user, new week	stored prefs, new week	Dashboard → Start	Prefill prefs?	Same as 1	n/a

			new week			
5	Mid-wizard reload	<code>wizardState.step < 9</code> stored	Wizard resumes	Skip edits?	Same as 1	Storage corrupt → restart wizard
6	Email deep-link share	<code>/share?planId=</code> exists	Plan Preview (read-only)	Fork plan	New wizard pre-filled	404 → not-found page

6 Core Page Specifications

6.1 Landing / ZIP-Gate

- **Desktop ≥ 768 px** 560 px centered card, hero tagline, animated gradient bg. “How It Works” 3-column below.
- **Mobile** Stack vertically, hero icon 48 px, input full-width, sticky **Get Started** CTA bottom.

6.2 Wizard (Hybrid Chat)

- Bubble chat stream 60 % width (desktop) / full (mobile).
- Bottom input area renders chips, steppers, file-drop as needed.
- Top-right stepper desktop; collapses to 100 % progress bar mobile.
- `aria-live="polite"` for bot messages.

6.3 Meal-Plan Review

- 2-column recipe cards (desktop) / single scroll (mobile).
- Show macros, cost, cook time, substitution badge.
- Fixed CTA bar bottom mobile.

6.4 Dashboard (Returning)

- **Desktop** Sidebar profile + week history; main pane "This Week" summary.
- **Mobile** Tab bar (Home | History | Settings).

6.5 Settings / Preferences

- Accordion sections, free-form “Anything else?” textarea.

6.6 Error / Empty States

- Illustrated ZIP unsupported, network-down, Instacart outage pages.

7 Form Audit & Enhancements

1. **Free-form context** textarea stored per-week.
2. **Week scope** toggle: “This week only” vs “Save as default”.
3. **Portion variability** checkbox *Kids only*, per-meal overrides.
4. **Cuisine mood** multi-tag (choose ≤ 3 weekly).
5. **Ingredient exclude** list separate from allergies.
Chatbot auto-asks follow-ups if free-form text missing required entities.

8 System Architecture

Layer	Tech	Notes
Frontend	Next.js 14 + Tailwind + shadcn	Firebase Hosting
Serverless	Firebase Functions (Node 20)	GPT & Instacart calls
Data	Firestore	users, mealPlans, instacartLists
AI	OpenAI GPT-4o-mini (chat + vision)	Function-calling

Email	Resend	Templates: <code>plan_confirmed</code> , <code>plan_failed</code> , etc.
Observability	Sentry + PostHog	Funnels & errors

InstacartGateway stub toggled by `featureFlag('noApi')` until keys live.

9 Functional & Non-Functional Requirements
9.1 Functional

ID	Requirement	Acceptance Test
FR-01	Unsupported ZIP reroutes to wait-list	100 % routing verified
FR-02	GPT meal-plan p95 < 5 s	Logged metrics
FR-03	Ingredient match fail < 10 %	Unit test
FR-04	Pantry dedupe ≥ 90 %	Integration test
FR-05	Resend email ≤ 30 s	Inbox QA
FR-06	Regenerate loop < 2 min	E2E test

9.2 Non-Functional

- LCP ≤ 2 s 4G.
- GPT round-trip p95 ≤ 7 s.
- Error budget < 1 % 5xx.
- WCAG 2.1 AA.

10 Claude Code Autopilot Work-Order

1. **Repo scaffold** `/apps/web`, `/packages/ui`, `/packages/core-ai`,
`/packages/core-flows`.

2. Guard-rails: TS strict, eslint-turbo, secrets-scan CI, branch `feat/mvp-wizard`.
 3. Implement:
 - `generateMealPlan()` OpenAI function.
 - `visionDetectPantry()` helper.
 - `InstacartGateway` stub switching to real.
 - Vitest flows for scenarios 1-6.
 4. Local setup: `pnpm i && pnpm dev` (mock Instacart).
-