

ROUTING & QUIZ LOGIC SPEC

Agent Type: Claude (Reasoning / Systems Logic)

Primary Objective

Design and implement the multi-step diagnostic quiz, server-side conditional routing, and deterministic recommendation engine that routes users to the correct problem, package, or product based on their inputs.

Core Responsibilities

- Build multi-step quiz UI (client-side) with validation.
- Implement server-side quiz submission endpoint.
- Compute primary pain, route_to path, and recommended_offer payload.
- Generate signed redirect token tied to lead_id.
- Ensure lead is created and tagged BEFORE redirect occurs.
- Return redirect URL + recommendation payload to frontend.

Quiz Input Fields (Required)

- Sub-niche (enum)
- Monthly revenue range
- Monthly ad spend range
- Primary traffic platform
- Primary pain (enum)
- Lifecycle maturity (email/SMS status)
- Commerce channel
- Name, Email, Phone, Company, Website
- UTM parameters + referrer

Routing Logic Rules

Routing must be computed server-side using deterministic rules. Client-side routing is not permitted for final redirects.

Example mapping: low_roas → /problems/low-roas
low_conversion_rate → /problems/low-conversion-rate
low_ltv_retention → /problems/low-ltv-retention
attribution_tracking → /problems/attribution-tracking
scaling_breaks → /problems/scaling-breaks
overreliance_on_paid → /problems/overreliance-on-paid

Recommendation Engine Output

The recommendation engine must output a structured JSON payload containing:

- offer_key
- offer_name
- price_display
- cta_type (checkout | book_call)
- rationale_bullets[]
- next_step_copy

Autonomy Rules

- Do not hardcode pricing; pull from config/constants.
- Do not modify UI or backend schemas.
- Pause and request clarification if routing conflicts arise.