

SAYSO | Product of ChitChat | CEO Briefing Summary

Onboarding Workflow Analysis (High-Level Summary)

Overview

The Sayso onboarding workflow is a guided, multi-step experience that collects a user's interests, subcategories, and deal-breakers to enable highly personalized recommendations. The system balances user experience, data integrity, and platform performance.

Workflow Steps

1. Landing Page (/onboarding)

Purpose: First-touch entry for new users.

Features:

- Welcome carousel
- “Get Started” → /register
- “Log In” → /login

No user data is collected here.

2. Registration / Login (/register, /login)

Purpose: Authenticate the user.

Flow:

- User provides email + password
- Must verify email before progressing
- After successful login → redirect to /interests

3. Interests Selection (/interests)

Purpose: Collect the user's high-level categories of interest.

Rules:

- Min: 3 interests, Max: 6

Storage:

- Saved temporarily in localStorage
- Not written to the database yet

Navigation:

- Moves forward via query parameters (/subcategories?interests=...)

4. Subcategory Selection (/subcategories)

Purpose: Capture more specific preferences within selected interests.

Rules:

- Min: 1 subcategory, Max: 10

Storage:

- localStorage + OnboardingContext state

Navigation:

- Passes interests + subcategories via URL to next step

5. Deal-Breakers (/deal-breakers)

Purpose: Final preference data.

Rules:

- Min: 1 deal-breaker, Max: 3

Critical Action:

- All onboarding data is saved atomically to the server
- Updates profile: onboarding_complete = true

Error Handling:

- If the save fails, the user still moves to the completion screen

6. Completion Page (/complete)

Purpose: Confirm successful onboarding.

Features:

- Confetti animation
- Auto-redirect to /home
- User profile now fully initialized

Data Flow Model

- OnboardingContext manages selections
- Temporary storage in localStorage enables recovery and back-navigation
- Final POST to /api/user/onboarding persists everything in one transaction using a Supabase RPC

Key Design Decisions

- Deferred persistence improves UX
- Atomic transaction ensures data consistency
- Graceful error handling maintains smooth flow
- URL-driven navigation supports direct linking

Security

- Authentication required for all steps
- RLS ensures users can only modify their own data
- Server-side validation ensures all stored data is clean

Enhancement Opportunities

- Add retry mechanism for save failures
- Improve offline support via service workers
- Add analytics for drop-off and engagement tracking

- Enable “resume onboarding” from server-side state