

YAZIO Trial Day Case Study

“Welcome Back Funnel: Personalization Experiment”

Estimated Time: ~4 hours (including preparation)

Focus: Product-minded frontend development, data fetching, A/B thinking, testing, and AI-assisted workflow

Overview

At Yazio, we continuously run experiments on our web funnel to improve conversion and re-engagement.

In this case study, you'll implement a mini onboarding “welcome back” funnel that uses **existing user data** and **fresh quiz inputs** to personalize the final experience — such as the paywall or CTA screen.

Your focus is to demonstrate how you combine **clean frontend implementation**, **data-driven personalization**, and **measurable experimentation** — not pixel-perfect visuals. Basic, clear UI is absolutely fine.

For inspiration, you can look at our public onboarding flow:

👉 yazio.com/app/onboarding/welcome

You're encouraged to use **AI tools and coding assistants** (e.g., ChatGPT, Cursor, Copilot, Claude, or any agents) to speed up development, explore ideas, or check your reasoning.

The key is to **balance automation and judgment** — use AI to help, but make the final decisions yourself.

The Challenge

Design and implement a lightweight **multi-step “Welcome Back” flow (4–5 screens)** that personalizes the final screen content based on a user’s stored profile and new quiz answers.

This flow is meant for users who are not Pro users yet but already answered a lot of questions in a previous funnel. Now they’re coming back and we want to try to convince them a second time.

Example Flow

1. **Welcome Back** – greet the user by name and invite them to continue.
2. **Confirm Goal** – prefill their goal (e.g., *Lose weight*), allow confirmation or change.
3. **Quiz Steps** – ask the user 1-3 more questions
4. **Summary** – show combined profile + new answers for confirmation.
5. **Paywall / CTA (A/B/C variants)**
 - Control (A): generic benefits, no personalization.
 - Variant B: personalized content based on the user’s goal (e.g., “Lose weight”).
 - Variant C: deeper personalization combining goal and preference (e.g., “Quick recipes for balanced meal plans”).

Data & Personalization

You can work with **mocked data sources** for:

- A user profile (e.g., name, goal, dietary preference).
- A simple mapping that links goals/preferences to personalized content (e.g., headlines, benefits).
- You can host or import your JSON data locally — no real API needed. Keep type safety in mind.

A/B Test Simulation

Simulate an experiment setup that differentiates between a **Control** and one or two **variants**.

Use deterministic group assignment based on the user's id so the same user consistently sees the same variant. Document your allocation strategy.

Track at least a few basic interaction events (console or mock util).
For example:

- *Screen views (per step)*
- *CTA clicks (with active variant)*

The goal is to demonstrate your ability to reason about **experiment design and measurable outcomes**, not to build a production analytics system.


Requirements

Your implementation should include:

- **A project built with Vue3 or React and TypeScript**
 - **Frameworks like Nuxt or Next.js are also fine**
 - **Feel free to use a starter template to speed up setup-time**
- **Data fetching and merging logic** for personalization
- **Simple State handling** (e.g. Pinia or composables) for user flow
- **A/B variant toggle** that affects the final screen
- **Automated testing** — ideally end-to-end (Playwright or Cypress)

Unit & Integration tests are welcome but not required — focus on E2E coverage of the core logic.

AI-assisted workflow: use any tool (ChatGPT, Cursor, Copilot, etc.) and reflect on how you balanced AI suggestions with your own decisions.

 **Design note:** Prioritize clarity, correctness, and maintainability over visual polish. A well-structured, readable solution is more valuable than a complex UI.



Deliverables

Please provide:

- A small project repository (GitHub, CodeSandbox, or zip).
 - Brief setup/run instructions.
 - A **5–10 minute walkthrough** covering:
 - Architecture and data flow
 - How you merged profile + quiz data
 - How the variant behaves and what you'd track
 - How you used AI, and where you intervened manually
 - What you'd improve or extend with more time
-



What We'll Evaluate

Dimension	What We're Looking For
Technical Craftsmanship	Clean, maintainable TypeScript and Vue/React structure (components, composables, state); thoughtful error handling.
Data & Personalization	Correct fetching, merging, and usage of personalized data to drive UI decisions.
Testing & Quality	Reliable end-to-end coverage of core flow and personalization; pragmatic, maintainable test design.
Experimentation Mindset	Clear Control vs Variant setup; measurable tracking; understanding of conversion metrics and A/B test rigor.
AI Fluency	Strategic use of AI tools; ability to critique, refine, and take ownership of AI-generated code.
Architecture & Scalability	Reasoning around structure, extensibility, and how this logic could scale in a larger app.
Communication	Clear explanation of reasoning, trade-offs, and collaboration with Design or Analytics.



Nice-to-haves (Optional Stretch Goals)

- Simple **feature-flag** helper

- Allowing the user to skip questions
 - Basic performance or accessibility consideration
 - Thoughts on **event schema or success metrics** you'd track in production
-

Time Guidelines

You'll work on your solution on your own, at your own pace — there's no hard time limit. We expect that about 4 hours should be enough to complete it.

Our goal isn't to judge the final result, but to understand how you think, make decisions, and approach the challenge.


During the Trial Day, we'll:

- Discuss your metrics and event design with Analytics/Product,
 - Pair on improvements with engineering,
 - Review your **testing approach and AI-assisted workflow**.
-

Evaluation Intent

This exercise is about how you **think, structure, and reason** — not how fancy your UI looks. We want to see how you:

- Translate product goals into maintainable, testable code.
 - Reason about user impact, data, and metrics.
 - Leverage AI tools effectively and responsibly.
 - Communicate your approach clearly and confidently.
-

 That's it — focus on clarity, reasoning, and data-driven decision-making. We care as much about *how* you think as *what* you build.