

StakeStack — an AI course builder, and a type system that cut runtime errors by 40%

Senior Frontend Developer · AI-powered course-creation platform · Oct 2024 – Feb 2026 · Remote (US)

40% ↓ RUNTIME ERRORS REDUCED	Manual + AI DUAL COURSE AUTHORIZING	End-to-end ARCHITECTURE TO DEPLOYMENT	Real-time ENGAGEMENT TRACKING
--	---	---	---

CONTEXT

StakeStack is an AI-powered course-creation platform that lets creators build, launch, and sell online courses faster. I led frontend development end-to-end — from architecture through deployment — on React, Next.js, and enterprise-grade TypeScript, building the core course builder and the shared type system the product runs on.

THE CHALLENGE

- Build a course builder flexible enough to support both hand-authored and AI-generated content in a single, coherent editing and publishing flow.
- Keep a fast-moving AI product reliable as it grew — evolving APIs and generated content made runtime type errors a constant risk.
- Raise engineering quality across the team while continuing to ship features.

APPROACH & KEY DECISIONS

A dual-authoring course builder

Built a full course builder that treats **manual authoring and AI-generated content as one flow** — creators can generate, then edit, publish, and track engagement without switching tools or mental models. The AI output drops into the same typed, editable structures as hand-written content.

Type-safety as a reliability strategy

Designed **strongly-typed APIs and shared TypeScript interfaces across frontend and backend**, so data shapes were guaranteed at compile time rather than discovered as production bugs. This **reduced runtime errors by 40%** — critical for a product where AI-generated data is inherently less predictable.

Advanced TypeScript as a team multiplier

Coached the team on advanced patterns — discriminated unions, utility types, custom hooks — lifting code quality across the codebase and making the platform safer for everyone to extend, not just me.

End-to-end ownership

Owned the frontend from architecture decisions through to deployment, keeping the system coherent as features and AI capabilities expanded.

WHAT I BUILT

- **Course builder** — unified manual + AI-generated authoring, with editing and publishing.

- **Real-time engagement tracking** — live insight into how learners interact with courses.
- **Shared, strongly-typed API layer** — interfaces spanning frontend and backend, the backbone of the 40% error reduction.
- **Team enablement** — advanced-TypeScript coaching and patterns adopted across the codebase.

Stack: React · Next.js (App Router) · TypeScript · Tailwind CSS · TanStack Query · AI content generation

OUTCOME

Delivered a dual-authoring AI course builder and a shared type system that **cut runtime errors by 40%**, keeping a fast-moving AI product reliable as it scaled. Raised team-wide TypeScript proficiency, making the codebase safer to extend across the whole team — over a 16-month engagement owning the frontend end to end.

WHAT THIS PROJECT SHOWS

Deep frontend architecture combined with real AI-product experience: designing type-safe systems that stay reliable as they scale, folding AI content generation into coherent product flows, and raising an entire team's engineering standard.

Richard Osuya · Senior Frontend Engineer & Team Lead

osuyarios@gmail.com · +234 810 077 9351 · stakestack.ai