

PRODUCT DESIGN CASE STUDY

Friendly Manager Website Builder

Replacing a 10-year-old website platform with a flexible, integrated builder for sports clubs.

ROLE	TIMELINE	SCOPE
Product Lead & Lead Product Designer	8 months	Research to launch

This was not just a redesign. It was a new product, a new service model, and a new source of recurring revenue.

Executive summary

Friendly Manager helps sports clubs manage registrations, teams, events, communications and payments. Its original website editor had been built more than a decade earlier and had become too limited for modern club needs.

I led the replacement project across research, product definition, UX/UI design, developer collaboration, testing, launch and commercial packaging. The result was a connected website platform that could serve complete beginners, experienced website users and Friendly Manager's internal custom-build team.

10+ years	8 months	10 customers	4 tiers
of legacy customer feedback	from discovery to beta	using it in week one	free to \$129/month

The business problem

The old platform allowed customers to replace text and images, but not reorganise pages, build new layouts or create a modern responsive site.

The design problem

The product had to be simple enough for volunteers with very little computer experience while still giving advanced users meaningful creative control.

The strategic opportunity

The strongest point of difference was integration with Friendly Manager, allowing club information to flow directly into a public website.

The commercial result

The builder created paid upgrade paths and enabled Friendly Manager to offer end-to-end custom website services.

1. Background and context

The original Friendly Manager website platform was intentionally simple. A customer opened a page, clicked edit, replaced an image or changed text, then saved. At the time it was built, this paint-by-numbers model reduced risk and made content updates approachable for non-technical club administrators.

Over the following decade, expectations changed. Club websites became more visual, mobile usage increased and customers wanted more control over their public presence. The same simplicity that had once been a strength became a limitation.

Customers could not:

- reorder page sections
- move content between areas
- create new page structures
- customise responsive behaviour
- apply modern animation or rich visual treatments
- build a website that reflected the quality of their organisation

The support history made the need visible. Friendly Manager had accumulated more than 10 years of requests describing what customers could not achieve. Replacing the platform was overdue, but the size of the problem meant it could not be treated as a cosmetic refresh.

2. My role and level of ownership

I was the project lead and final product decision-maker. In practice, I worked across Product Lead, Product Owner and Lead Product Designer responsibilities.

My responsibilities included:

- reviewing historic customer feedback and support requests
- interviewing club administrators across New Zealand and Australia
- analysing established website-building products
- defining the product vision, scope and requirements
- creating wireframes, prototypes and high-fidelity interface designs
- documenting expected behaviour for development
- reviewing implementation quality and resolving design gaps
- tracking delivery and defects in ClickUp
- coordinating internal, beta and real-device testing
- helping define plans, permissions and upgrade paths
- supporting launch and prioritising ongoing improvements

The delivery team was deliberately small: one primary developer, one backend developer, a subject matter expert and a group of internal testers. A key developer left during the project, so I also helped preserve the product direction through a handover and onboarding process for the replacement developer.

3. Discovery and customer research

The project began with an unusually rich research base. We already had years of support tickets showing recurring pain points, but I did not rely on historic requests alone. I worked with club administrators across New Zealand and Australia to understand what they expected from a modern website and which problems were most important to solve.

Customers asked for familiar website features such as video, richer graphics, more flexible layout tools and better mobile control. However, the strongest insight was that the website should not behave like a separate product.

The website needed to be connected to club operations.

General-purpose tools such as Wix, Webflow, WordPress, Elementor, Weebly and Joomla were useful references. I had around 15 years of experience building websites with these kinds of products, which helped me understand established interaction patterns and the trade-offs between simplicity and control.

The competitive analysis clarified the strategic position: Friendly Manager did not need to beat general-purpose builders at every feature. It needed to be the best website platform for clubs already running their organisation through Friendly Manager.

4. Product strategy

The product strategy had three connected parts: simplify the experience for beginners, unlock deeper control for advanced users, and integrate the website with the core club-management platform.

4.1 Progressive levels of control

A single editor would either overwhelm beginners or frustrate experienced users. I designed a progressive model with different levels of access.

Level	Designed for	Core capability
Basic	Low-confidence users	Edit text and images inside a professionally built, responsive theme.
Intermediate	Confident content managers	Resize, reorder and move

		content with greater layout control.
Premium	Advanced website users	Create rows, columns and blocks, with detailed responsive overrides.

The key design principle was progressive disclosure: provide a safe, clear default, then reveal more control only when the user needed it.

4.2 Integration as the differentiator

The most important product decision was to connect the website builder directly to Friendly Manager data and workflows. Clubs could publish and maintain teams, registrations, player information, fixtures, results, events, news, sponsors and calendars without manually recreating the same information in another system.

Event publishing became a strong example. A club administrator could create an event in Friendly Manager, click a button and automatically generate the required website page, rows, blocks and content structure. This reduced duplicate work and meant users did not need to understand how to construct an event page from scratch.

5. UX architecture and key design decisions

5.1 Visual entry point instead of a page list

Our initial assumption was that customers would want to land on a list of pages. That structure made sense from a content-management perspective, but usability testing showed that it did not match the mental model of many club administrators.

Users wanted to see their actual website, identify the thing they wanted to change and edit it in context. They were not thinking about information architecture. They were thinking, “I can see it, so let me change it.”

I changed the landing experience from a page-management view to a visual website view. This was one of the most important improvements for making the product approachable to less experienced users.

5.2 Global properties

To create consistency without asking customers to style every page, I designed global properties for logos, colours, fonts and shared visual settings. A change to a global property flowed across the website automatically.

Local overrides were still possible, preserving flexibility for advanced users while keeping the default experience coherent.

5.3 Reusable blocks

Reusable blocks allowed commonly repeated content to be created once and used across the site. Examples included sponsor strips, registration calls to action, contact information and promotional sections. This reduced duplicate effort and supported consistent design across large sites.

5.4 Responsive behaviour

Themes, rows and blocks were designed to be responsive by default, so basic users did not need to understand breakpoints. More advanced users could switch between desktop, tablet and mobile views and override individual properties, including dimensions, visibility, spacing and layout.

This created a layered system: safe automatic behaviour at the base, with detailed control available when required.

5.5 Animation without unnecessary complexity

Animation highlighted the wider challenge of feature richness. Exposing every animation option would create an intimidating interface. I therefore designed a simple mode with a small number of clear choices and an advanced mode for users who wanted deeper control.

6. Theme management and design systems

Theme management was the most structurally complex design area. The user experience needed to explain how global settings, page-level settings, reusable components and local overrides worked together.

The system needed to answer several questions clearly:

- What changes everywhere?
- What changes only this page?
- What changes only this block?
- What happens when the theme changes?
- Which controls are available on each plan?
- How can advanced users customise without making the product unsafe for beginners?

The solution was a layered design system. Global properties created brand consistency. Themes provided responsive foundations. Reusable blocks created repeatable content patterns. Local

overrides preserved creative freedom. Subscription levels determined how much of this system a customer could access.

7. Design and delivery process

I began in Figma, producing wireframes, prototypes and high-fidelity designs. I reviewed early direction with the subject matter expert, then translated the product into specifications and build documentation for the developers.

Delivery was tracked in ClickUp. As features were completed, I first reviewed them against the design and functional intent. They then moved through internal testing, wider team testing and customer beta testing.

Typical review sequence

1. Design and functional review by me
2. Internal testing team
3. Wider company testing
4. Customer beta testing
5. Real-device checks on phones and tablets
6. Fix, retest and release

The distributed development setup introduced communication and implementation challenges. The delivered interface did not always match the Figma work precisely. I responded by prioritising the core customer experience, documenting design gaps and creating a roadmap for ongoing UX refinement rather than blocking the entire release.

8. AI-assisted design and quality assurance

The project happened during a period when AI tools became substantially more capable. My process evolved while the product was being built.

8.1 Extending the interface system

Once I had established the visual language and interaction patterns, I trained Claude to help produce additional interface components based on that direction. This reduced repetitive design effort and helped me move faster from a defined pattern to a broader system.

The developers also began using AI in their workflow, allowing us to collaborate around faster prototyping and implementation.

8.2 Reusable QA skill

Template QA was another significant time cost. A manual review required checking padding, margins, column gaps and responsive behaviour across desktop, tablet and mobile.

I created a reusable Claude skill to help inspect websites for these consistency issues. The automated check reduced repetitive review time, but it did not replace human judgment. Finished sites were still tested by real people using real phones and tablets.

This approach is best described as AI-assisted design operations: AI handled repeatable checks and pattern application, while people remained responsible for usability, visual quality and final approval.

8.3 Future direction

The longer-term product direction is to allow a club to generate much of a website from a small number of prompts. The theme, block and integration architecture built during this project creates a foundation for that future workflow.

9. Testing and iteration

The testing group deliberately included people at both ends of the experience spectrum: customers who had barely used website software and customers who were comfortable with professional design tools.

The internal team was also useful because it contained a broad range of digital confidence. Observing people use the product revealed issues that would not have appeared through design review alone.

The strongest example was the landing experience, but testing also reinforced the need to simplify dense controls, rely on responsive defaults and avoid exposing advanced settings before they were needed.

10. Pricing, packaging and commercial model

The product was designed with a progression from low-risk entry to advanced control. The pricing structure included a free version, followed by plans at \$39, \$79 and \$129 per month.

This was more than feature gating. The experience supported the commercial model. Customers could begin with a safe, simple theme, gain confidence and then move into more advanced tools as their needs grew.

Custom website service

The builder also enabled Friendly Manager to offer a complete custom website service, including stakeholder consultation, information architecture, UX and visual design, site construction, content setup, training and handover.

This opened a new service-based revenue stream and created additional work inside the company. The product therefore supported both self-service recurring revenue and higher-value professional services.

11. Customer example: Tauranga City Football

Tauranga City Football previously had an outdated website that no longer reflected the quality or scale of the club. The new platform allowed us to rebuild it as a modern, manageable and season-aware site.

My contribution included:

- restructuring the homepage and navigation
- improving the journey through registrations, teams and seasonal information
- creating AI-assisted imagery
- producing video content
- designing the information architecture
- building the site with the new platform
- ensuring the club could manage ongoing updates independently

Sports club content changes throughout the year. A homepage may need to move from registrations to fixtures, finals, holiday programmes or prizegiving. To support this, the builder included scheduled content blocks. The club could decide what appeared, when it became visible and when another block replaced it.

The outcome was not only a better-looking site. It was a website that could follow the operational rhythm of the club and remain manageable by the customer.

12. Launch setback and response

The most serious setback occurred around publishing and domain management. The product was ready, but the domain infrastructure failed when the team attempted to launch a site.

I worked with the developers through the issue while the domain server was rebuilt. Because the problem was discovered during beta, customer impact was limited and there was no significant public delay.

The incident expanded our definition of product quality. Testing a website builder could not stop at editing flows and responsive layouts. It also needed to cover the full path from configuration to a reliable live domain.

The release process therefore needed stronger end-to-end testing around publishing, domain configuration, infrastructure and fallback scenarios.

13. Outcomes and early impact

The product is still relatively new, so the available evidence is early rather than long-term. Even so, the first results demonstrated clear customer and business value.

Early adoption: Around 10 customers began using the builder in the first week, with very little formal marketing or launch communication.

Customer independence: Clubs could make meaningful website changes without relying on a developer for every update.

Product expansion: Friendly Manager could offer a website experience that competed more credibly with modern website platforms.

Recurring revenue: The free-to-paid model created a natural path into higher subscription tiers.

Service revenue: The custom website offer created a new, higher-value revenue stream.

Operational efficiency: Integrated content and automatic page generation reduced duplicate administrative work.

14. What I would do differently

The project delivered a capable product, but it used more of a big-bang approach than I would choose today. The larger initial scope created complexity and meant parts of the UI and UX needed to be simplified or redesigned after testing.

If I started again, I would:

- use AI from the beginning
- build a working front end earlier instead of relying as heavily on static Figma designs
- release a smaller MVP sooner
- set clear expectations that the first version was intentionally limited
- test with customers earlier and more frequently
- bring development closer to the design process
- validate publishing and domain infrastructure much earlier

The central lesson was to shorten the distance between customer problem and working product. A smaller release would have reduced risk, created faster learning and avoided some redesign work.

15. What this project demonstrates

I am most proud of taking a product the company had needed to replace for more than a decade and leading it through to a real, usable platform.

The project demonstrates my ability to connect:

- customer research and product strategy
- complex systems and approachable user experiences
- visual design and commercial packaging
- design intent and technical delivery
- AI-assisted speed and human quality control
- long-term product vision and practical MVP trade-offs

I did not only design an interface. I defined the product direction, coordinated a small team, worked through delivery and infrastructure setbacks, changed the experience when evidence contradicted our assumptions, and helped turn the final platform into a new commercial offering.

The result was a connected website platform that served beginners, advanced users and Friendly Manager's own website team, while creating a foundation for future AI-generated club websites.

Project summary

Problem

A restrictive 10-year-old editor could no longer support modern club websites.

Approach

Research historic feedback, interview customers, benchmark established tools and design a progressive, integrated builder.

My role

Lead product strategy, UX/UI design, requirements, delivery review, testing,

Key decision

pricing and launch decisions.

Differentiator

Design for progressive control rather than forcing beginners and experts into the same interface.

Outcome

Connect website content directly with Friendly Manager workflows and data.

Learning

A modern platform, new upgrade revenue, a custom-build service and early customer adoption.

Release a smaller MVP earlier and use AI-assisted code prototyping from the beginning.