CASE STUDY 2: CHEVS

EXECUTIVE SUMMARY: Chevs consultancy focused on designing and developing a secure, scalable, and accessible website for CHEVS, a youth-led feminist organisation working across West Africa in a high-risk environment for LGBTQI+ communities. The project required balancing visibility and storytelling with strong digital safety considerations, ensuring that the platform could support advocacy work without exposing the organisation or its community to unnecessary risk. The website was delivered through a phased approach. Phase I focused on rapid design and deployment of a high-performance frontend, while Phase 2 centres on building a robust, multilingual content infrastructure using a headless CMS. Across both phases, the work prioritised security-conscious architecture, modular design, and maintainability. The result is a platform that functions not only as a communications tool, but as intentional digital infrastructure supporting CHEVS' work while embedding safeguards appropriate to its operating context.

WHAT MAKES THIS CASE STUDY AN EXAMPLE:

Building for a queer-led organisation in Africa means every technical decision carries real weight. This case study shows how we navigated that - balancing a tight deadline, a security-first architecture, multilingual content needs and a design process that had to get the identity right before anything else could move forward.

Approach:

C4AC Labs' approach combined complex problem-solving across design and engineering to ensure the platform was built to serve CHEVS' mission without compromise.

Security was treated as a first principle rather than a feature - shaping infrastructure decisions from the ground up. Design was grounded in audience insight, with close collaboration ensuring the final product maintained brand identity, user experience standards and visual consistency across every page.

The CMS architecture was built with content accessibility, inclusion and diversity at its core, giving CHEVS full editorial autonomy across both English and French without technical dependency.
Flexibility was built into every layer - from the component-based design system to the headless content architecture - ensuring the platform could scale and adapt as the organisation grows. Critical incident management and deployment continuity were factored into the CI/CD pipeline from day one, so the team could respond quickly when it mattered.

The result is a platform built for a truly global approach - multilingual, secure, and designed to serve a diverse audience across multiple geographies and contexts.

Objectives:

End-to-end development of a scalable, high-performance website architecture, integrating front-end and back-end systems to ensure reliability under load and optimal user experience.
Designed and implemented a modular, multilingual content infrastructure (English/French) using a headless CMS (Sanity), enabling independent content management at scale and supporting non-linear editorial workflows.

Architected a headless CMS solution to address complex information security considerations, reducing exposure of sensitive data and decoupling content management from the presentation layer.

Embedded security-first principles across the development lifecycle, including secure third-party integrations (e.g. Brevo), minimising attack surfaces and mitigating risks to data exposure and external vulnerabilities.

Led UI/UX design and front-end implementation, translating brand identity into a responsive, accessible interface aligned with modern HTML, CSS, and JavaScript standards.
Coordinated cross-functional workflows across design, front-end, and back-end development to deliver Phase I within a rapid deployment timeline (10 days post-design sign-off), reflecting agile delivery practices.

Implemented structured content models and reusable components to support maintainability, scalability, and future feature expansion.

Ensured maintainability and long-term extensibility through clear system architecture, documentation, and alignment with modern development workflows (version control, iterative deployment readiness).

Ensured accessibility considerations were integrated into UI design and front-end implementation, aligning with WCAG standards for inclusive user experience.

IMPLEMENTATION

Phase I — Design & Initial Build

UI/UX Design

- The website was designed in Figma, with multiple revision cycles in close collaboration with the CHEVS team. This process focused on ensuring that the visual language, tone, and user experience felt authentic to the organisation and the communities it represents.
- A component-based design system was adopted to ensure consistency across pages and allow the platform to scale more easily in later phases.

Frontend Development

- The frontend was built using Next.js and React with TypeScript, using server-side rendering (SSR) for performance and SEO optimisation
- Tailwind CSS was used for styling, with responsive layouts built to ensure usability across devices and varying network conditions
- All priority pages were designed, built and deployed within 10 days of final design sign-off, requiring tight coordination between design and development

Infrastructure & Deployment

- The site was deployed via Vercel, with DNS configured through Namecheap, routing through Vercel's edge network
- Version control was managed through GitHub using a feature branch strategy, with all changes reviewed before being merged
- CI/CD pipeline managed through GitHub Actions — automating build and deployment processes on every merge to main, ensuring only reviewed and approved code reached production

Integrations

- Brevo integrated via REST API to handle newsletter sign-ups and grant form submissions
- Form validation handled client-side in React; submission logic managed server-side through Next.js API routes, ensuring sensitive contact data was never exposed on the frontend

Debugging & Issue Resolution

- Given the short delivery timeline, rapid debugging was critical, key issues resolved during the sprint included cross-device responsiveness and DNS configuration challenges during deployment
- All fixes were handled through the feature branch workflow, reviewed and merged to main before going live

Phase 2 — CMS Migration & Platform Extension (In Progress)

CMS Architecture

- Migrating hardcoded content to Sanity, a headless API-first CMS
- Content fetched using GROQ — S anity's native query language — via Sanity's JavaScript client
- Sanity studio configured with a bilingual content architecture, with English and French managed as parallel document schemas using Sanity's internationalisation patterns
- Role-based access controls implemented within Sanity studio, ensuring editors operate within defined permission boundaries without access to infrastructure or configuration

Security Architecture

- Headless architecture deliberately chosen to decouple the content layer from the frontend, reducing attack surface and limiting exposure of sensitive organisational data
- Access to backend systems restricted through role-based permissions across both Sanity and GitHub

Integrations

- PayPal payment portal integrated via PayPal's REST API
- Webhook handling implemented through Next.js API routes to manage donation confirmation and error states securely

Debugging & Issue Resolution

- Ongoing debugging of CMS migration covers content query optimisation in GROQ, language-switching logic and Sanity schema validation
- All fixes tracked and resolved through the same feature branch and GitHub Actions pipeline established in Phase I