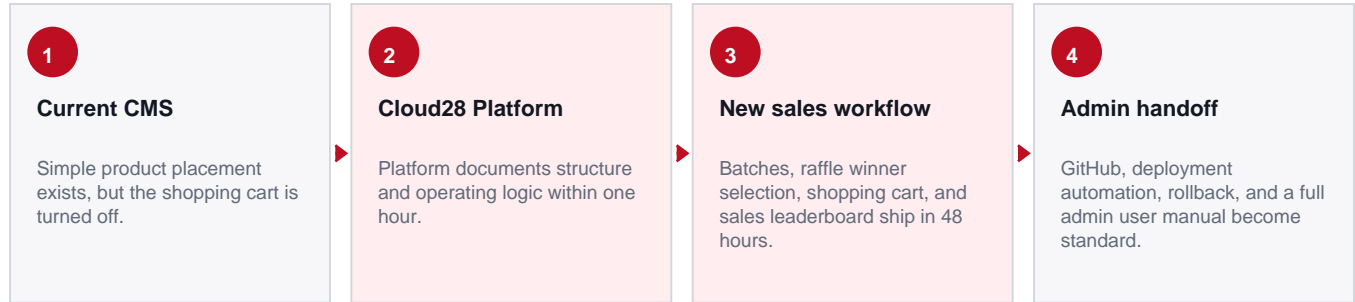


From Product Display CMS Commerce Platform

How Cloud28 turned a difficult-to-update CMS website into a documented, version-controlled web app with new batch sales, raffle winner selection, shopping cart activation, sales leaderboard, deployment controls, rollback capability, and a full admin user manual in three days.



CLIENT	Client Website
WORKFLOW	Current state: simple CMS product placement with shopping cart turned off. New state: Cloud28-created batch sales, raffle winner selection, sales leaderboard, team competition, and controlled production release flow.
SCALE	Transaction
ENGAGEMENT	Cloud28 platform discovery, documentation, feature development, deployment automation, GitHub version control, database backup, rollback capability, and admin user manual delivery.
DEPLOYED	Zero-to-one knowledge transfer completed in 3 days: website documented within 1 hour and new feature development completed within 48 hours.
OUTCOME	The business moved from vendor-dependent maintenance to a controlled application lifecycle with visible releases, recovery paths, and a full admin manual for training and daily usage.

01 Overview / Executive Summary

Client Website needed a full website revamp after inheriting a CMS website platform from an existing source CMS site. The source CMS site was usable, but the team did not have documentation that explained how to safely manage the site, change product behavior, manage database state, or extend the application without relying on trial-and-error discovery.

The operational problem was not only design polish. Client Website needed Cloud28 to create a new operating workflow: add product batches, activate cart-based product, activate raffle eligibility from sales activity, select winners, publish the result, and show a sales leaderboard for team competition.

Cloud28 approached the engagement as a zero-to-one knowledge transfer. The platform documented the website structure within one hour, then used that to complete new feature development within 48 hours. By the end of the three-day engagement, the client website had the new business functionality, the operating controls needed to keep improving the application, and a user manual for admin training and daily usage.

The result is a practical foundation for future web operations: GitHub-backed version control, repeatable production deployment, database backup, rollback capability, admin operating documentation, and a clear path for new product and sales workflows. The outcome compresses work that can easily stretch across weeks into a controlled, three-day implementation cycle.

<p>1 KNOWLEDGE TRANSFER</p> <p><1 hour</p> <p>Website structure documented quickly enough for immediate engineering action.</p>	<p>2 FEATURE DELIVERY</p> <p><48 hours</p> <p>Batch sales, raffle, cart activation, and leaderboard capabilities completed.</p>	<p>3 ADMIN HANDOFF</p> <p>Manual ready</p> <p>Full admin user manual delivered for training and usage.</p>
--	--	--

02 Current Operation

Before the Cloud28 engagement, the website operated as a simple CMS presence. Products could be placed on the website, but the shopping cart had been turned off. That meant the site could show products, but it was not running as an active campaign commerce system.

The inherited source CMS CMS platform also lacked practical documentation for the client website team. When the business wanted to move beyond product display into batch selling, raffle-based promotion, and team competition, and shopping cart checkout, development first required discovery of how the current website was structured and how it could be safely changed.

In the current state, there was no connected workflow for creating a batch, turning on buying behavior, tracking sales leadership, running a raffle, selecting a winner, or announcing campaign results. Those capabilities were the new workflow Cloud28 was asked to create, not the baseline operation.

Step	Manual action today	Operational friction
Product placement	Team places products inside a simple CMS website.	The site can display product information, but does not operate as a live campaign sales workflow.
Shopping cart	Shopping cart is turned off.	Visitors cannot move through a complete website purchase flow.
Batch selling	No product batch workflow exists in the current CMS operation.	The business cannot easily run bounded sales campaigns from the website.
Raffle program	No built-in raffle eligibility, winner selection, or winner announcement workflow exists.	Promotions require separate manual work or cannot be run confidently from transaction activity.
Sales competition	No sales leaderboard exists for team competition.	Managers and sales leaders lack a live view of campaign performance by team or seller.
Production release	Deployment, database backup, and rollback are not treated as one repeatable controlled process.	A failed change can become a production recovery problem.

A DOCUMENTATION GAP

Static CMS

Existing platform displayed products but operating logic was not documented.

B CHANGE RISK

Manual release

Production updates lacked formal version control, backup, and rollback discipline.

C REVENUE DELAY

Cart off

The website displayed products but did not yet support active cart-based campaigns.

03 Solution

Cloud28 converted the inherited CMS into a known operating environment. The first milestone was documentation: mapping application structure, update paths, core data objects, deployment behavior, and the places where new capabilities could safely attach.

With that knowledge base in place, Cloud28 created the requested commerce workflow. Product batches give the business campaign-level control. Shopping cart activation lets customers buy through the website. Raffle logic connects eligible sales to winner selection and winner announcement. The sales leaderboard turns the campaign into a visible team competition.

The new workflow changes the website from a product display CMS into an operating sales platform. Sales leaders can track campaign performance, teams can compete around visible results, and management can use the website itself as the source of truth for batches, entries, winners, and release history.

Cloud28 Platform acts as the full-stack service layer for this workflow, leveraging agentic engineering to build and maintain connector code, schema definitions, validation rules, release scripts, database backup routines, rollback steps, and repeatable QA checks.

Capability	What it does	What changes for Client Website
Website documentation	Maps CMS structure, code paths, content rules, database touchpoints, and release behavior.	The team can make future changes from known system knowledge instead of vendor-dependent discovery.
Product batches	Adds structured selling batches to existing products, including campaign boundaries and operational status.	Product launches can be managed as controlled sales campaigns instead of static product placement.
Shopping cart activation	Turns the disabled shopping cart back into an active purchase path.	The website can support live customer buying behavior during a campaign.
Raffle workflow	Creates raffle eligibility, winner selection, and winner announcement from campaign activity.	Promotions become traceable, easier to run, and easier to explain to customers.
Sales leaderboard	Creates a visible leaderboard for sales leaders, teams, and campaign competition.	The sales team gets direct performance feedback and a reason to compete around live results.
GitHub and deployment control	Introduces true version control, automated production deployment, database backup, and rollback paths.	Every release has source history, deploy repeatability, and recovery capability.

04 Realization Plan

The engagement was intentionally compressed. Cloud28 first made the unknown system legible, then used that documentation to deliver features and operational controls. This sequencing matters: fast development is only valuable when the team also gains the knowledge and recovery paths required to operate the software afterward.

Day / Phase	Focus	Customer-readable deliverable	Acceptance signal
Hour 0-1	Discovery and documentation	Website structure, update paths, data objects, and deployment behavior documented.	Team can identify where changes should be made and how the website is assembled.
Day 1	Architecture and release baseline	GitHub repository, source-control workflow, deployment automation plan, backup and rollback approach.	A production change can be traced to source and paired with a recovery path.
Day 1-2	Feature build	Product batches, shopping cart activation, raffle winner selection, winner announcement, and sales leaderboard.	New campaign workflow works against existing product and website structures.
Day 3	Production hardening	Deployment automation, database backup routine, rollback checklist, admin user manual, and knowledge transfer.	The client can train admins, release, verify, and recover without undocumented vendor dependency.

S SPEED

3-day cycle

Knowledge transfer plus feature delivery plus production controls.

R

Weeks avoided

Typical undocumented CMS discovery and redevelopment can easily take weeks.

C CONTROL

Rollback ready

Production deploys are paired with backup and recovery discipline.

05 Effectiveness Over Time

Effectiveness should be measured as the website moves from simple CMS product placement to an actively managed campaign commerce platform. The baseline is the current state: shopping cart turned off, no batch sales workflow, no raffle winner workflow, no sales leaderboard, unclear release history, and limited rollback confidence. The pilot state proves that the new Cloud28 workflow operates correctly. Production measures stability, campaign usability, and traceability.

Measure	Baseline	Pilot	Production target	Optimization signal
Website knowledge	Undocumented CMS behavior.	Core application map completed.	Change requests begin from documented paths.	New features update the documentation set as part of delivery.
Feature cycle time	Discovery-first; can stretch into weeks.	New campaign features completed within 48 hours.	Small features move through defined build, review, deploy, and backup steps.	Lead time trends down without increasing rollback incidents.
Shopping cart	Cart is turned off.	Cart activation tested against product and batch flow.	Customers can move from product interest to website purchase flow.	Checkout drop-off and campaign conversion can be measured.
Sales leaderboard	No team competition leaderboard exists.	Sales leader and team performance visible in the workflow.	Teams can compete around live campaign results.	Reporting expands by batch, seller, team, product, and campaign.
Raffle trust	No built-in raffle eligibility, winner selection, or announcement flow.	Eligibility, winner selection, and announcement tested.	Winner announcement is generated from traceable campaign context.	Audit checks confirm no missing or duplicate eligibility records.
Release control	Limited source history, backup, and rollback visibility.	GitHub and deployment automation introduced.	Each release has source history, database backup, and rollback plan.	Release notes and recovery tests become routine.

[Redacted]

[Redacted]

[Redacted] tracking,

[Redacted]

M ADMIN ENABLEMENT

Manual ready

Full user manual delivered for admin training and daily website usage.

R RECOVERY

Rollback path

Production releases are paired with database backup and recovery steps.

G VERSION CONTROL

GitHub source

Application changes are tracked through a true source-control workflow.

The most important control is visibility. Cloud28 leaves the website in a state where source code, configuration, release steps, database recovery, and admin usage are no longer hidden behind an undocumented CMS arrangement. GitHub becomes the source of truth for application history and change review, while the admin user manual becomes the training reference for daily operation.

Production deployment is automated so releases can be repeated consistently. Before meaningful database-changing releases, backup steps protect the current state. Rollback steps give the team a defined response if a release introduces an unexpected behavior.

Exceptions should remain explicit. Batch setup issues, shopping cart activation problems, raffle eligibility anomalies, duplicate winner candidates, or incomplete leaderboard attribution should be surfaced for human review instead of silently passing through the workflow.

This operating model turns the application foundation where ~~CLIENT~~ website

more than a one-time website fix. It creates a maintainable web can train admins, use the new campaign features, add future features, recover from mistakes with confidence.