

BONUS CASE

What if, instead of just managing a website,
I built a content system from the ground up?

MIACHIU-TRAVEL

Building a Content System, Not Just a Travel Website

Overview

Building a Content System Not Just a Travel Website

AI-Assisted / Vibe Coding × System Design

This side project started during my working holiday in Canada. When an international friend asked for Taiwan travel recommendations, I began to wonder: instead of simply sharing a Google Maps Saved List, could I build a travel guide that would be easier to update and share?

Starting with 480 personal Google Maps saves, I used AI-assisted / Vibe Coding to explore the site architecture, data flows, and deployment approach in collaboration with AI, applying my previous experience in content management and operations to a real-world need.

I independently planned and made decisions across product positioning, content architecture, CMS, data flows, and multilingual functionality.

System Design

AI Collaboration

Architecture Decisions

Challenge

Data Volume

480 Google Maps saves created a large-scale content migration challenge.

Data Import

480 entries could not be created manually.

Cost Constraints

The system had to balance functionality, content quality, and maintenance cost.

Budget Constraints

Tools, hosting, and maintenance had to remain low-cost.

Maintenance Barrier

Content updates needed to be manageable without relying on development resources.

Stability

Scalability

Content maintenance should not depend on development.

Three constraints shaped the architecture — batch processing, low-barrier maintenance, and reduced dependence on development.

Approach

Batch Content Management

Google Maps → Google Sheet → Contentful

Exported the Google Maps Saved List, organized the data in Google Sheets, then used Apps Script to batch-create and update Contentful entries.

Low-Cost Multilingual Setup

Apps Script LanguageApp → Contentful

Translated content before import, keeping human-written English as the source and automating only the less-used languages.

Low-Barrier Content Maintenance

Contentful + Google Sheet

Separated technical operations from content editing, allowing non-technical users to update content through familiar tools without touching the codebase.

Frontend Architecture

Next.js + Tailwind CSS

Used reusable components for categories, tags, bilingual content, and responsive layouts, making new content easier to add without changing the core architecture.

01

Google Maps Saved List

Data Source

02

Google Sheets

Data Prep

03

Apps Script

Data Sync

Auto Translation

04

Contentful

CMS

05

Next.js

Frontend

06

Cloudflare Pages

Deployment

Approach

From Manual Entry to Batch Processing + Semi-Automated Publishing

● PHASE 01 | Build the MVP

Launch First, Validate Later

Turned the concept into a functional website, establishing the visual direction, content structure, CMS integration, and mobile experience.

Next.js

Tailwind CSS

Cloudflare Pages

Contentful

● PHASE 02 | Fix Translation Issues

Frontend Translation Hit Limits

Free translation services triggered request limits and frontend loops, making English translation unreliable. Batch processing and request deduplication stabilized the workflow.

Google Translate API

● PHASE 03 | Build a Better Workflow

Move Translation Upstream

Moved batch import and English translation into a Google Sheets → Apps Script → Contentful pipeline, keeping the frontend translation API as a fallback.

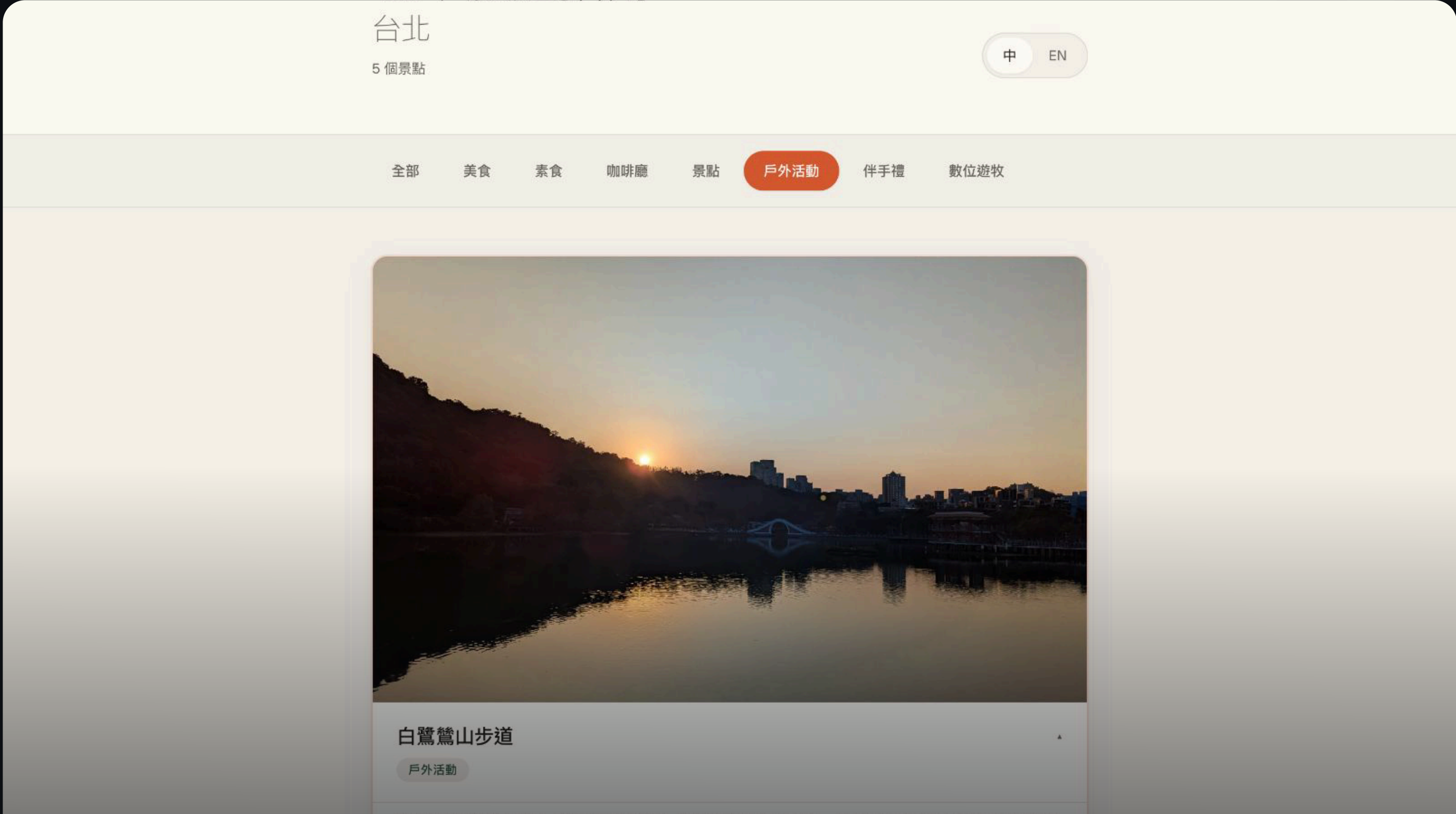
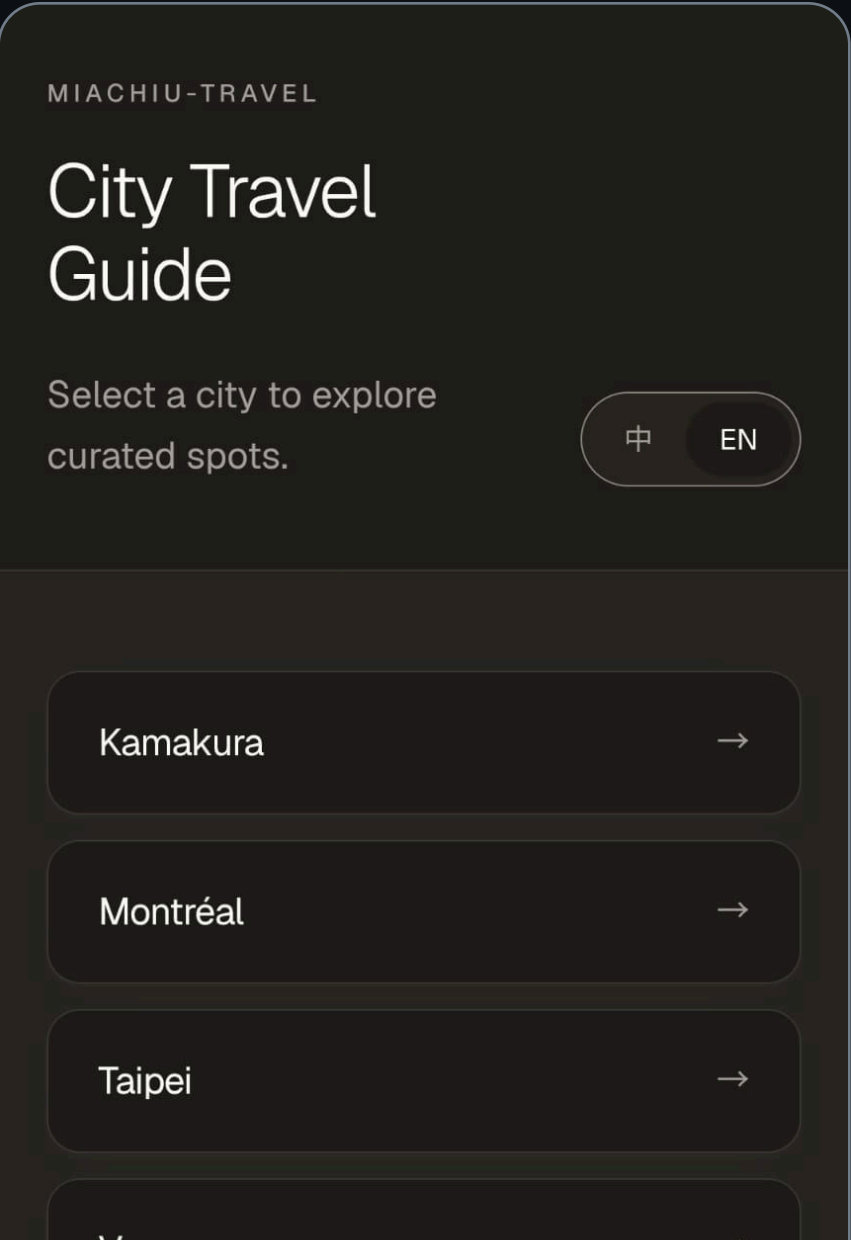
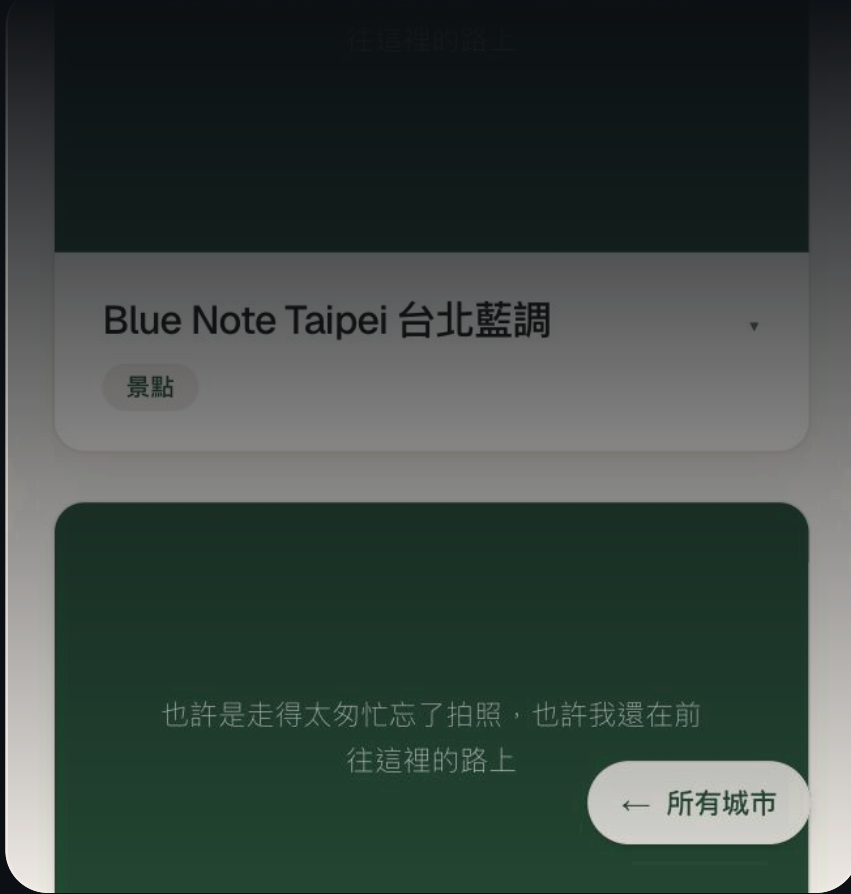
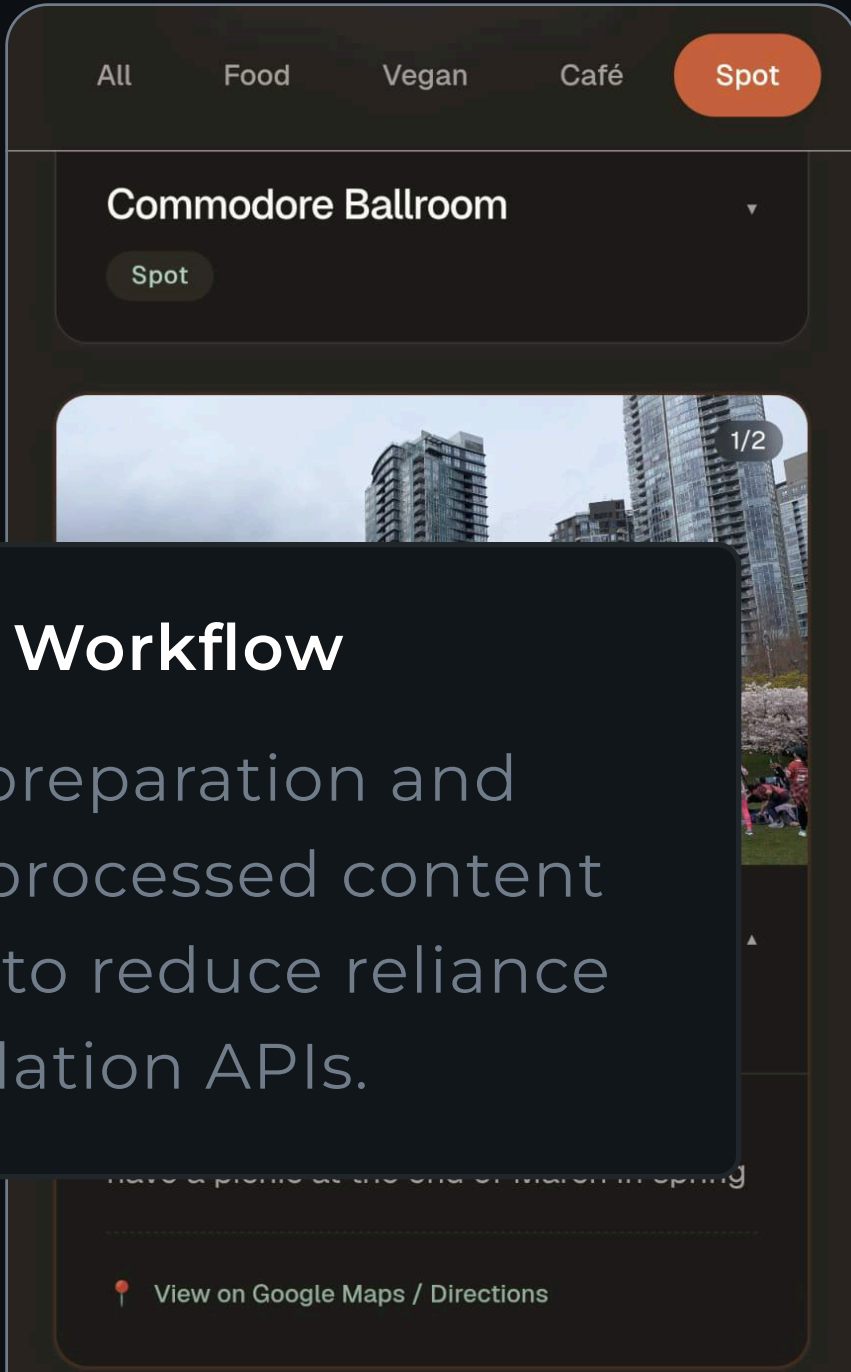
Google Sheet

Apps Script

Contentful

———— **BONUS CASE · SELF-INITIATED PROJECT** - Building a Content System, Not Just a Travel Website

From a Personal Website to a **Maintainable Content System**



01 Efficiency & Scalability

Turned content into a batch-based workflow, reducing repetitive work and speeding up city and spot publishing.

02 Non-Technical Maintenance

Separated content operations from code, allowing updates and translations through familiar tools without developer support.

03 Semi-Automated Workflow

Automated data preparation and translation, then processed content through the CMS to reduce reliance on frontend translation APIs.