

NEXT.JS · PAYLOAD CMS · 6 МОБ

# Міграція сайту на Next.js (App Router) + Payload CMS

Чиста типізована архітектура, шість мов із зручним керуванням перекладами та адмінка, у якій контент-менеджер працює сам — без розробника.

**з 2014**

full-cycle studio

**10+**

років у digital

**400%**

середній ROI

**CA · US**

клієнти

Next.js App Router

Payload CMS

TypeScript

Tailwind CSS

i18n · hreflang

ISR · RSC

## ЧОМУ ЦЕ ЗВЕРНЕННЯ КОРОТКЕ

ТЗ у вас зібране грамотно — вказаний стек, App Router, Payload, рівень володіння TypeScript і навіть критерій «адмінка має бути зручною для регулярного самостійного наповнення». Тож не лию воду про «команду професіоналів». Краще покажу, як ми робимо саме таку міграцію — по кожному вашому пункту.

Ось у чому суть: міграція на Next.js + Payload — це не «перенести верстку». Це переукладання даних у типізовані колекції, налаштування локалізації на рівні полів і маршрутизації під шість мов так, щоб і Google, і контент-менеджер почувалися добре.

## Що ви написали → як ми робимо

Ваш пункт	Наш підхід
Перенести сайт на Next.js (App Router)	App Router як основа (Server Components за замовчуванням, Client — лише де потрібна інтерактивність). Pages Router я б тут не радив — на мультимовному проєкті з ISR App Router дає чистіший роутинг і кеш. Контент і медіа переносимо міграційними скриптами, а не руками — щоб нічого не загубилось.
6 мов + зручне керування перекладами	Локалізація Payload на рівні полів: одна сутність — переклади поруч, перемикач локалі прямо в редакторі. Контент-менеджер бачить, де переклад ще порожній, і не лазить по різних записах.
Адмінка для самостійного наповнення (статті, переклади, медіа)	Налаштовуємо колекції під ваш реальний контент, а не дефолтні поля: зрозумілі назви, групування, прев'ю медіа, чернетки та публікація. Мета — щоб людина без розробника додала статтю за пару хвилин.
Чиста й масштабована архітектура	Строга типізація end-to-end (типи Payload → фронт без «апу»), модульна структура, передбачуваний шар даних. Проєкт, який ваша команда зможе розвивати далі, а не боятись зачепити.
Базове SEO + коректна мультимовна маршрутизація	Локалі в URL (/uk, /de...), hreflang без дублів, canonical, мета й Open Graph на кожну локаль, окремі sitemap. Плюс ISR — свіжий контент без повного ребілда.

Сайт у відгуку не вказаний — тож не вигадую «проблеми» наосліп. Коли дасте доступ, перший крок у нас завжди — інвентаризація поточного контенту й типів сторінок, щоб міграція пройшла без втрат.

## Шість мов без болю для контент-менеджера

### Переклади поруч, а не в окремих записах

У Payload locale — це не копія сайту, а поле. Стаття одна, мови перемикаються в редакторі. Це прибирає головний жах мультимовності — «де загубився переклад» і «чому дві версії роз'їхались».

### Чесно про fallback

Домовляємось заздалегідь: якщо локаль ще не заповнена — показуємо мову-резерв чи ховаємо сторінку? Це маленьке рішення, але саме воно рятує від «порожніх» сторінок німецькою у видачі.

- Перемикач мов, який не ламає поточний шлях користувача.
- Дати, числа й формати — під локаль, не «захардкожені».
- Інтерфейс адмінки теж локалізуємо, якщо з контентом працюватимуть різномовні редактори.

## SEO, ПРО ЯКЕ ЧАСТО ЗАБУВАЮТЬ ПРИ МІГРАЦІЇ

### Щоб переїзд не з'їв позиції

Міграція — момент, де легко втратити трафік. Тому SEO ми закладаємо в архітектуру, а не «прикручуємо» в кінці.

- **hreflang без дублів** — кожна локаль знає про інші, Google не плутає версії.
- **301 зі старих URL** — якщо структура змінюється, ваги сторінок не зникають у нікуди.
- **Мета й OG на кожну мову** — не один шаблон на всіх, а локалізовані заголовки й описи.
- **Core Web Vitals** — RSC + грамотний кеш дають швидкість «з коробки», а не ціною плагінів.

*Це не повний SEO-аудит — він окремо. Тут — базова коректність, яку ви й просили в ТЗ, зроблена правильно з першого дня.*

## Фази роботи

---

### 1 Інвентаризація та модель даних

Розбираємо поточний сайт: типи сторінок, контент, медіа, мови. Проектуємо колекції Payload під ваш реальний контент.

---

### 2 Каркас Next.js + Payload

App Router, типізований шар даних, локалі, базовий деплой. Скелет, на якому вже видно структуру.

---

### 3 Міграція контенту й медіа

Перенесення даних скриптами, налаштування 6 локалей, перевірка що нічого не загубилось.

---

### 4 Адмінка під редактора

Доводимо панель до стану «зручно наповнювати самому»: поля, групування, прев'ю, чернетки/публікація.

---

### 5 SEO, маршрутизація, передача

hreflang, sitemap, 301, мета по локалях. Прогін, документація, передача команді з можливістю розвивати далі.

*Фази, не календар: точні строки узгоджуємо після інвентаризації — коли видно реальний обсяг контенту й мов. Обіцяти терміни до перегляду сайту було б нечесно.*

---

ПРО ДОСВІД — ЧЕСНО

## Чому ми

Next.js, headless-CMS і мультимовні проекти — наша робоча щоденність, а не разовий експеримент. Payload CMS — той самий клас headless-рішень: code-first схема, типізація, локалізація на рівні полів — ми працюємо в цій моделі.

Чесно про портфоліо: значна частина наших проектів — під NDA, тож публічно показуємо лише вибірку. Нижче — кейси, близькі за інженерною суттю: каталоги, міграції, високонавантажені платформи. Якщо потрібні приклади саме під ваш стек — покажемо в чаті релевантні роботи окремо.

## WebCoreLab — full-cycle digital studio · Toronto, Canada

Дизайн, розробка, SEO/GEO та DevOps під одним дахом — від ідеї до продакшну й підтримки. На ринку з 2014, 10+ років у digital, 20+ років сукупного досвіду команди.

Архітектурний підхід, вимірюваний результат, повний контроль на боці власника.

### ПАРТНЕРСТВА ТА ВИЗНАННЯ

● Clutch · Top Agency Canada

● DesignRush · Best SEO

● G2 · High Performer

● Google Partner

● HubSpot Partner

● Cloudflare Partner

● Semrush Partner

● ISO 27001 Aligned

## Релевантні кейси

Близькі за інженерною суттю проекти — міграції, каталоги, e-commerce і платформи з високим навантаженням.

E-COMMERCE · CATALOG

### Massage Warehouse

Великий каталог, чистий UX, швидкий checkout.

→ конверсія каталогу вгору

ENTERPRISE · PLATFORM

### XPO Logistics

Високонавантажена платформа, стабільність під навантаженням.

→ масштаб + надійність

PLATFORM · SCALE

### LatinAmericanCupid

Платформа на мільйони користувачів.

→ масштабованість

SAAS · GEO

### TechFlow

B2B SaaS, гео-видимість і органіки.

→ зростання у ніші

E-COMM · MULTI-REGION

### GreenLeaf

Каталог + лідген по регіонах.

→ регіональні заявки

HEALTHCARE · CATALOG

### MedCore

Структурований каталог послуг.

→ органіки + довіра

Усі кейси на [webcorelab.com](https://webcorelab.com) →

WebCoreLab · Full-cycle digital studio · Toronto, Canada · з 2014  
[webcorelab.com](https://webcorelab.com) · [case-studies](https://webcorelab.com/case-studies)