

CMS Content Model & Field Checklist (v1.0)

Asset Amacı: Kurumsal web siteleri, otel portalları ve B2B projelerinde CMS içerik modelini (Content Type, Field, Relation) hatasız kurgulamak için hazırlanmıştır. Temel hedef; içerik giriş sürecini standardize etmek, Next.js veri akışına uyum sağlamak ve SEO metadatasını (schema/meta) otomatize ederek editoryal operasyonu hızlandırmaktır.

Kim Kullanır?: İçerik Liderleri, Ürün Sahipleri (Product Owner), Teknik Liderler ve SEO Uzmanları.

A) Ölçüm & Önceliklendirme Checklist'i

CMS mimarinizin sağlamlığını aşağıdaki 10 kritik maddede denetleyin:

- ☐ **Envanter Kontrolü:** Tüm içerik türleri (Oda, Paket, Blog, SSS, Hizmet) tanımlandı mı?
- ☐ **Content Type Ayrımı:** Sayfa şablonları ile veri tipleri (Type) birbirinden net ayrıldı mı?
- ☐ **Zoranlı Alanlar (Core Fields):** Title, Slug, Summary, Body ve Cover Image her type için standart mı?
- ☐ **SEO/Meta Katmanı:** Canonical, Meta Title/Desc ve Social Image alanları her içerikte mevcut mu?
- ☐ **İlişkisel Mimari (Relations):** Odalar ile Paketler, Bloglar ile Hizmetler arası bağlar kuruldu mu?
- ☐ **Çok Dil (Localization):** Locale bazlı slug yönetimi ve hreflang tetikleyicileri planlandı mı?
- ☐ **Editoryal Akış (Workflow):** Taslak (Draft), Önizleme (Preview) ve Yayında (Published) süreçleri net mi?
- ☐ **Schema Yapısı:** SSS (FAQ), Makale veya Hizmet şemaları için gerekli alanlar tanımlı mı?
- ☐ **Versiyonlama:** Modelde yapılacak güncellemeler için bir v1.0 / v1.1 geçiş planı var mı?
- ☐ **Yıllık Bakım (365):** İçerik modelinin performansını ölçmek için yıllık denetim takvimi oluşturuldu mu?

B) Problem → Kök Neden → Çözüm Tablosu

Problem	Kök Neden	Çözüm
Kopya İçerik / URL Karmaşası	Eksik Slug ve Hreflang yönetimi	CMS seviyesinde benzersiz (unique) slug zorunluluğu getir.
Geliştirici Zorluğu	Düzensiz/Karmaşık Field isimleri	CamelCase veya Snake_case standart bir naming convention kullan.
SEO Kayıpları	Meta alanlarının opsiyonel kalması	Meta alanlarını "Zorunlu Field" yap ve karakter sınırı koy.

C) 14 Günlük CMS Model Sprint Planı

Veri modelini temizleme ve Next.js uyumlu hale getirme takvimi:

- Gün 1–4:** Mevcut içerik audit'i, Content Type setinin netleştirilmesi ve çekirdek field standartlarının belirlenmesi.
- Gün 5–7:** İlişkisel modelleme (Relation Map) ve çok dilli yapı kurallarının dökümantasyonu.
- Gün 8–11:** Editoryal rollerin tanımlanması, Preview workflow kurgusu ve Next.js veri mapping testleri.
- Gün 12–14:** Örnek içerik tipleriyle son testler, editoryal eğitim ve v1.1 backlog'un oluşturulması.

D) Öncesi / Sonrası KPI Tablosu

Metrik	Önce (Plansız)	Sonra (Model Odaklı)
İçerik Girme Süresi	Yüksek / Karmaşık	Düşük / Akıcı
API Yanıt Uyumu	Hatalı / Eksik Veri	%100 Uyumlu JSON

SEO Hata Oranı	Sıkça Unutulan Metalar	Sıfır Hata (Zorunlu Alan)
----------------	------------------------	---------------------------

E) Deliverables (Teslim Edilecekler)

- Content Type Kataloğu:** Tüm içerik türlerinin teknik şeması.
- Field Set Standardı:** Çekirdek ve opsiyonel alanların listesi.
- Relation (İlişki) Haritası:** İçeriklerin birbirine nasıl bağlanacağını gösteren diyagram.
- SEO & Schema Sözleşmesi:** Editoryal ekipler için rehber döküman.