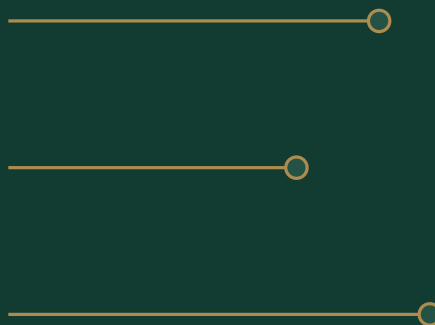


WORDPRESS PRO

Wydajność, bezpieczeństwo i SEO z pomocą AI



PROJEKT TWORZY CZŁOWIEK.
AI WSPIERA PROCES.

06

10 ROZDZIAŁÓW / JEDEN SYSTEM DEMONSTRACYJNY

Spis treści

Wprowadzenie i mapa projektu	3
01 Audyt, który prowadzi do decyzji	4
Brief audytu i definicja sukcesu · Mapa systemu i granice zaufania · Ryzyko i uczciwy język dowodów	
02 Wydajność widziana przez użytkownika	7
Core Web Vitals bez jednej magicznej liczby · Pomiar terenowy i laboratorium · Obraz hero i stabilne proporcje	
03 Cache bez naruszania prywatności	10
Co wolno przechować w cache · Unieważnienie po zmianie treści · Kontrola nagłówków bez udawania HTTP	
04 Zapytania i interakcje pod kontrolą	13
N+1 jest hipotezą do sprawdzenia · Paginacja i object cache · INP: mniej pracy po kliknięciu	
05 Bezpieczeństwo jako zestaw granic	16
Role, capabilities i najmniejsze prawa · Aktualizacje, zależności i sekrety · REST, nonce i publiczne uploads	
06 Kopia, odtworzenie i bezpieczne zmiany	19
Kopia to komplet spójnych danych · Odtworzenie w izolacji · Staging, wdrożenie i rollback	
07 Techniczne SEO bez sprzecznych sygnałów	22
Adresy, canonical i statusy · Robots, noindex i ochrona stagingu · Sitemap i linki wewnętrzne	
08 Dane strukturalne i treść zgodna z faktami	25
Organization i Service opisują fakty · Walidacja treści i źródła prawdy · Treści AI przechodzą przez draft	
09 Monitoring, logi i koszt AI	28
Monitoring zaczyna się od pytań · Logi bez prywatnych pytań · Koszt AI, kolejka i wynik nieznany	
10 Raport końcowy i odbiór WP Demo Lab	31
Pełny raport z generatora · Plan napraw i porównanie przed/po · Odbiór, testy i dalsza praca	
Źródła techniczne i dalsza lektura	34

Kliknij rozdział. Nazwa tomu w stopce prowadzi do spisu. Pełne pliki techniczne i instrukcje odtworzenia są w EXAMPLES.

Od obietnicy do dowodu

Ten tom zamyka serię praktycznym audytem fikcyjnego WP Demo Lab. Uczy porównywać obserwacje, ryzyko i koszt naprawy oraz oddzielać pomiar od przypuszczenia. AI pomaga przygotować decyzję, a człowiek odpowiada za odbiór.

Wcześniejsze etapy dają wspólny system: motyw blokowy prezentuje firmę, wtyczka realizacji dostarcza dynamiczną listę, formularz uruchamia przygotowanie szkicu, biblioteka porządkuje materiały, a asystent wyszukuje wiedzę z kontrolą dostępu. Tutaj sprawdzamy ich granice razem. Szybka strona nie może ujawnić prywatnej odpowiedzi przez cache; poprawne SEO nie może zamienić stagingu w publiczne źródło; automatyzacja nie może ominąć praw redaktora.

Materiał można przejść samodzielnie. Folder EXAMPLES zawiera kompletną kartę fikcyjnych ustaleń, generator raportu w Pythonie, oczekiwany raport, testy i runbook odtworzenia. Python działa bez WordPress i bez połączenia z modelem. Dzięki temu sprawdzisz kolejność ryzyk, kompletność dowodów i strukturę dokumentu przed przygotowaniem własnego środowiska. To ćwiczenie audytowe, a nie instalator pięciu wcześniejszych projektów.

Wartości i obserwacje demonstracyjne mają etykietę illustrative. Dla Core Web Vitals pozostawiamy brak pomiaru; nie podajemy wymyślnego wyniku PageSpeed ani oceny bezpieczeństwa z nieuruchomionego skanera. Pole planned opisuje cel naprawy, nie jej wykonanie. Własne pomiary dopiszesz dopiero po zapisaniu profilu, zakresu, źródła i dat. Raport ma jasno powiedzieć również to, czego jeszcze nie wiadomo.

- Rezultat: raport ryzyk, plan napraw, procedura pomiaru i kryteria odbioru.
- Warunek startu: Python 3.11+ dla ćwiczenia; osobny staging do weryfikacji WordPress.
- Granica AI: propozycje, przegląd i testy lokalne; bez samodzielnej publikacji lub zmian w produkcji.

Brief audytu i definicja sukcesu

Najpierw zapisz, jakie decyzje ma umożliwić audyt. Dla WP Demo Lab sukces oznacza działający publiczny serwis, chronione przepływy prywatne oraz naprawy, których efekt można sprawdzić i w razie potrzeby cofnąć.

Firma demonstracyjna prezentuje usługi i realizacje, udostępnia publiczne materiały oraz pozwala uprawnionym osobom pytać o wiedzę. Audyt obejmuje pięć stron i dwa prywatne endpointy. HOME ma szybko pokazać ofertę; Usługi prowadzą do kontaktu; Realizacje mają paginowaną listę; biblioteka pokazuje aktualną wersję pliku; intake zapisuje szkic. Te cele określają, które opóźnienia i błędy naprawdę blokują pracę.

Ustal środowisko: domena, wersje rdzenia i PHP, hosting, cache, lista aktywnych wtyczek, sposób wdrożenia i właściciele. Profil kontraktu wymaga funkcji dostępnych od WordPress 6.6 i PHP 8.2, lecz własny staging powinien używać aktualnych wspieranych i załatanych wydań. Brak dokładnego numeru wersji w demonstracji nie jest dowodem, że środowisko klienta jest zgodne. Numer zapisuje osoba wykonująca test.

Kryteria sukcesu ustal przed poprawkami. Dla prywatnego ask: właściwe ACL i brak współdzielonego cache w sukcesie oraz błędzie. Dla intake: draft, trwałe rozpoznanie powtórzenia i brak publikowania. Dla strony: brak regresji funkcji, poprawne adresy i zebrane dane wydajności. Cel CWV jest mierzalny dopiero po uzyskaniu reprezentatywnych danych terenowych. Sam wynik laboratoryjny nie zamyka tego warunku.

- Właściciel audytu akceptuje zakres, wyjątki oraz sposób gromadzenia dowodów.
- Każde ustalenie ma dowód, wpływ, plan, test odbioru i osobę odpowiedzialną.
- Poza zakresem demonstracji: prawdziwy klient, prawdziwe skany i naprawy produkcyjne.

01 / ARCHITEKTURA

Mapa systemu i granice zaufania

Rozdziel publiczne renderowanie, zapis szkiców i prywatne wyszukiwanie wiedzy. Audyt śledzi przepływ danych między nimi, bo najważniejsze błędy często powstają na połączeniach usług, a nie w pojedynczym pliku.

Motyw wp-demo-lab odpowiada za wygląd, nie za prawa domenowe. Wtyczka wp-demo-projects publikuje realizacje jako wdl_project i blok wp-demo-lab/projects. Wtyczka wp-demo-library przechowuje publiczne wdl_material, temat wdl_topic i wersję dokumentu. Oddzielne funkcje ograniczają skutki zmiany motywu. Czytelnik może odtworzyć ten układ na stagingu, ale nie powinien aktywować dwóch kopii tej samej wtyczki z różnych folderów serii.

Formularz prowadzi przez walidację i adapter AI do POST intake. Dedykowana integracja ma capability wdl_ingest, a wynik jest szkicem. Ask wymaga zalogowania i read_wdl_knowledge; WordPress ustala grupy, gateway wywołuje prywatną usługę Python. W profilu lokalnym używa adresu 127.0.0.1:8787. Produkcja wymaga właściwej prywatnej sieci i TLS. Sekret WDL_RAG_TOKEN istnieje wyłącznie po stronie serwerów, poza repo i przeglądarką.

Zaznacz miejsca trwałego przechowywania: bazę WordPress, publiczne uploads, prywatne źródła poza webroot, indeks RAG oraz rekordy idempotencji. Draft rekordu nie chroni bezpośredniego URL pliku w uploads. Własny audyt ma ustalić, czy dane są publiczne z decyzji redaktora, czy przypadkiem dostępne z infrastruktury. To rozróżnienie wpływa na backup, cache, logi i procedurę usunięcia dokumentu.

Granice przepływu

01 Przeglądarka

Publiczne HTML lub sesja z capability.

02 WordPress

Motyw, CPT, intake draft i gateway ask.

03 Usługa prywatna

ACL przed rankingiem; sekret serwerowy.

04 Magazyny

Publiczne uploads oraz oddzielne źródła prywatne.

01 / PRIORYTETY

Ryzyko i uczciwy język dowodów

Dobry raport oddziela obserwację od interpretacji. W demonstracji wszystkie ustalenia są syntetyczne; w realnym audycie każde musi wskazać plik, żądanie albo eksport, który inna osoba może bezpiecznie odtworzyć.

Karta audit-fixture.json zawiera dziesięć typowych problemów, od publicznego cache ask do wymyślonych ocen w JSON-LD. Oceniamy wpływ i prawdopodobieństwo umownie w skali od jeden do pięć. Iloczyn porządkuje pracę: P1 od szesnastu, P2 od dziewięciu, reszta P3. Wynik nie jest statystycznym prawdopodobieństwem włamania. Ujawnienie danych może wymagać natychmiastowej eskalacji niezależnie od pozycji w tabeli.

Trzy etykiety wystarczą do pilnowania prawdy. Illustrative oznacza materiał do ćwiczenia; measured oznacza wynik dostarczony z opisem źródła; planned oznacza zmianę lub efekt oczekiwany. Generator nie weryfikuje autentyczności eksportu. Wpisanie measured nie zastępuje podpisu wykonawcy ani dowodu. Nie mieszaj prognozy skrócenia czasu z wynikiem po wdrożeniu. Porównanie przed/po musi obejmować także brak wykonanej naprawy.

PROMPT DO CODEXA / Codex: przegląd kompletności audytu

Wklej do Codexa wraz z wymienionymi plikami. Oczekuj propozycji do przeglądu; wykonanie zmian wymaga własnego stagingu.

Przeczytaj audit-fixture.json, README i wspólny kontrakt. Zbuduj tabelę id, dowód, hipoteza, ryzyko, właściciel i brakujący test. Wskaż sprzeczności między prawami intake, ACL ask i cache. Nie pobieraj adresów, nie zmieniaj konfiguracji, nie skanuj strony. Oddziel dane syntetyczne od rzeczywistych i nie dodawaj żadnych wyników PageSpeed. Zaproponuj kolejność napraw jako uzasadniony plan z zależnościami.

KONTROLA: Odbiór: każde twierdzenie prowadzi do istniejącego ustalenia; brak dowodu pozostaje brakiem. Priorytety mają uzasadnienie, a cel nie jest opisany jako uzyskany wynik.

Core Web Vitals bez jednej magicznej liczby

Wydajność dotyczy pojawienia się treści, reakcji na działanie i stabilności układu. Core Web Vitals opisują te trzy doświadczenia oddzielnie. Ocena jednej strony nie zastępuje oceny ważnych szablonów i grup urządzeń.

LCP wskazuje pojawienie się największego elementu treści w obszarze widocznym. INP opisuje reakcję strony na interakcje, a CLS przesunięcia układu. Dla dobrego doświadczenia progi wynoszą odpowiednio do 2500 milisekund, do 200 milisekund i do 0,1. Ocena opiera się na 75. percentylu, osobno dla urządzeń mobilnych i komputerów. Wszystkie trzy muszą spełnić warunek good, aby mówić o zaliczeniu zestawu.

Średnia ukrywa wolniejszą część wizyt. Przykładowo szybkie komputery mogą obniżyć średnią czasu ładowania, choć znaczna grupa telefonów nadal czeka. Percentyl opisuje granicę rozkładu, a nie średnią z wyników. Nie podstawiaj też laboratoryjnego TBT za INP; to inna metryka diagnostyczna. Dla małego stagingu często nie ma danych terenowych. Uczciwy status brzmi wtedy unmeasured, nawet gdy test laboratoryjny wygląda obiecująco.

Generator przyjmuje już zagregowane p75 i nie wylicza ich z surowych zdarzeń. Nie obiecuje zgodności z raportem dostawcy RUM lub CrUX. Źródło, okres, zakres i liczba próbek są obowiązkowe dla danych measured. Liczba próbek jest kontekstem do oceny, nie automatycznym dowodem reprezentatywności. Jeśli brakuje któregoś pola, raport nie może udawać pełnego pomiaru. Dokumentuj również wyłączone widoki i ograniczenia zbierania.

Metryka	Good	Poor
LCP	≤ 2500 ms	> 4000 ms
INP	≤ 200 ms	> 500 ms
CLS	≤ 0,1	> 0,25

02 / METODA

Pomiar terenowy i laboratorium

Laboratorium służy diagnozie przy powtarzalnych warunkach. Dane terenowe pokazują doświadczenie rzeczywistych wizyt. Potrzebujesz obu perspektyw, lecz nie wolno łączyć ich w jeden pozornie precyzyjny wynik ani ukrywać ich ograniczeń.

Zapisz URL, typ urządzenia, szerokość, profil sieci, obciążenie procesora, stan cache i wersję narzędzia. Powtarzaj test w tym samym układzie; porównuj rozkład wyników zamiast wybierać najładniejszy przebieg. Test zimny i ciepły odpowiadają na różne pytania. Rozszerzenia przeglądarki, zalogowanie oraz narzędzia debugowania mogą zmienić wynik, dlatego profil testu jest częścią dowodu. W tej książce takiego testu WordPress nie wykonano.

Na własnej witrynie dane terenowe zbieraj zgodnie z przyjętą polityką prywatności. Nie przysyłaj pełnych adresów z parametrami użytkownika ani treści formularzy. Ustal grupy szablonów, mobile i desktop, okres oraz zasady filtracji. Dla prywatnych ekranów brak raportu publicznego nie oznacza dobrego doświadczenia. Jeśli stosujesz własne RUM, opisz metodę zbierania i agregacji. Dostępność danych w narzędziu publicznym zależy także od wystarczającego ruchu.

Od pomiaru do decyzji

01 Profil

Zakres, urządzenie, sieć, cache i wersje.

02 Laboratorium

Powtarzalne ślady i diagnoza przyczyny.

03 Dane terenowe

p75 i kontekst prawdziwych wizyt.

04 Decyzja

Potwierdzone ustalenie lub jawny brak danych.

W ćwiczeniu najpierw wygeneruj raport z pustym CWV, potem obejrzyj test ręcznych danych measured. Wartości z testu są danymi testowymi, nie wynikiem WP Demo Lab. Przenosząc format do własnego audytu, użyj rzeczywistego eksportu i zachowaj jego identyfikator. Raport powinien dać się odtworzyć bez domyślania się, który wynik dotyczył telefonu, a który komputera.

02 / OBRAZY

Obraz hero i stabilne proporcje

Największy obraz bywa kluczowym elementem LCP, ale najpierw potwierdź to w śladzie. Poprawny rozmiar pliku, szybkie odkrycie zasobu i miejsce zarezerwowane w układzie rozwiązują różne części problemu.

Dla hero nad pierwszym ekranem nie stosuj lazy loading. Jeśli pomiar potwierdzi jego znaczenie, rozważ wysoki priorytet pobierania. Nie nadawaj go wszystkim obrazkom: zasoby konkurują o połączenie i procesor. Obrazy niżej mogą korzystać z loading=lazy. Width i height opisują proporcje; CSS może nadal dopasować szerokość. Sprawdź sizes względem rzeczywistego layoutu, bo błędna deklaracja potrafi wymusić znacznie większy wariant srcset.

KOD W PRAKTYCE / HTML / Fragment HTML hero — dopasuj do motywu

```

```

PLIK: image-markup.html

Fragment ilustracyjny; pliki obrazów i działający WordPress nie są dołączone.

W produkcji wykorzystaj rozmiary generowane przez WordPress i sprawdź finalny HTML, nie tylko ustawienia edytora. Na telefonie obejrzyj currentSrc, wymiar wyświetlany, transfer i jakość. Okładki materiałów oraz miejsca na osadzone elementy powinny mieć stabilne proporcje. Zmiana obrazu może poprawić transfer, ale nie naprawi wolnego HTML ani blokującego skryptu. Odbiór obejmuje wygląd, dostępny opis i porównanie śladu w identycznym profilu.

Co wolno przechować w cache

Cache przyspiesza ponowne użycie danych, lecz wymaga odpowiedzi na pytanie, komu wolno zobaczyć ten sam wynik. Publiczna strona firmy i odpowiedź asystenta zależna od uprawnień należą do różnych klas.

Publiczne HTML anonima można objąć cache strony lub CDN po ustaleniu warunków. Odpowiedź zależna od sesji, koszyka, prawa dostępu albo danych formularza nie powinna trafić do wspólnego magazynu. Dla ask i intake przyjmujemy private, no-store oraz jawny bypass warstw cache. Private zabrania współdzielenia; no-store instruuje, by nie przechowywać odpowiedzi. Sam no-cache pozwala na przechowanie z wymaganą rewalidacją i nie spełnia kontraktu prywatnego RAG.

Sprawdź żądanie i odpowiedź. Nagłówek Authorization, cookie logowania, metoda inna niż GET/HEAD lub Set-Cookie mogą wymagać bypass. Nie cacheuj całego REST według jednej reguły. Publiczne opublikowane realizacje różnią się od prywatnego ask. Dostawca CDN może mieć własne pierwszeństwo reguł, warianty klucza i wyjątków. Dlatego przykładowa polityka jest pseudokonfiguracją: nie wolno wkleić jej jako działającego pliku Nginx ani Apache.

PROMPT DO CODEXA / Codex: mapa cache i danych

Wklej do Codexa wraz z wymienionymi plikami. Oczekuj propozycji do przeglądu; wykonanie zmian wymaga własnego stagingu.

Przeanalizuj kontrakt oraz headers-policy-example.config. Zbuduj macierz publiczne HTML, publiczne CPT, ask, intake, odpowiedzi błędów, sesje i Authorization. Dla każdego opisz magazyn, warunek bypass, nagłówki i test z dwiema sesjami. Uwzględnij ścieżkę REST z query string oraz rest_route. Nie zmieniaj serwera i nie deklaruj działającego CDN. Wskaż reguły wymagające adaptacji u dostawcy.

KONTROLA: Odbiór: prywatne odpowiedzi nigdy nie mają celu shared-cache; błędy nie omijają polityki. Zgodność nagłówków jest oddzielona od faktycznego zachowania infrastruktury.

03 / AKTUALNOŚĆ

Unieważnienie po zmianie treści

Cache musi wiedzieć, kiedy wynik przestaje być prawdziwy. Publikacja realizacji, nowa wersja materiału i zmiana uprawnień mają różne skutki. Długie TTL bez procedury unieważniania potrafi ukryć poprawne wdrożenie.

Po aktualizacji realizacji odśwież jej stronę i publiczną listę, a przy zmianie menu także powiązane widoki. Nie opieraj procedury wyłącznie na ręcznym przycisku w panelu. Zapisz zdarzenie, zestaw kluczy lub tagów oraz dowód purge. Obserwuj jednocześnie cache WordPress, object cache, reverse proxy i CDN: usunięcie jednego wpisu nie gwarantuje odświeżenia wszystkich warstw. W audycie zaznacz właściciela każdej z nich.

Nowa wersja wdl_material powinna wycofać stare fragmenty z indeksu. Dokument ma identyfikator, wersję, grupę, lokalizator i stan active. Cache wyszukiwania, jeśli w ogóle występuje, nie może ignorować wersji ani ACL. W naszym kontrakcie odpowiedzi ask są no-store. Prywatny wewnętrzny cache obliczeń wymaga osobnego projektu z kluczem uwzględniającym uprawnienia i wersję oraz z procedurą usunięcia. Nie dodawaj go jako przypadkowej optymalizacji.

Zmiana materiału i odbiór

01 Publikacja wersji

Nowy numer i zatwierdzona treść.

02 Wycofanie starej

Nieaktywne stare fragmenty oraz zależne klucze.

03 Odbudowa indeksu

Tylko dozwolone aktualne źródła.

04 Test odczytu

Zwrócone ID i wersje zgodne z ACL.

Przećwicz również usunięcie dokumentu i odebranie grupie dostępu. Wynik nie powinien powrócić z cache po chwili ani po restarcie usługi. Test na stagingu ma zapisać identyfikatory, bez kopiowania prywatnej treści do raportu. Jeśli nie można ustalić, kto unieważnia dany magazyn, raport powinien wskazać tę lukę jako problem operacyjny.

03 / TEST OFFLINE

Kontrola nagłówków bez udawania HTTP

Pakiet dostarcza rzeczywisty sprawdzacz dostarczonych nagłówków. To test logiki polityki, a nie test serwera. Dzięki negatywnym przypadkom łatwo zobaczyć, jakie konfiguracje naruszają kontrakt i czego sam checker nie dowodzi.

Funkcja `check_private_response` normalizuje nazwy nagłówków i rozdziela dyrektywy `Cache-Control`. Wymaga `private` oraz `no-store`, odrzuca `public` i `s-maxage`, a także sygnał `HIT`. Plik `http-cases.json` zawiera sukces, odmowy i błędy usługi. Nie wysyła żądań do `example.test`. Wynik pozytywny potwierdza interpretację karty testowej, nie odseparowanie danych dwóch rzeczywistych użytkowników. Te dwa poziomy odbioru mają osobne protokoły.

KOD W PRAKTYCE / PYTHON / Uruchomienie kontrolera lokalnego

```
import json
from pathlib import Path
from audit_report import check_private_response

cases = json.loads(
    Path("http-cases.json").read_text()
)
for case in cases:
    errors = check_private_response(case)
    print(case["id"], errors)
```

PLIK: `check_headers.py`

Wykonywalny helper offline; używa fikcyjnych kart HTTP, nie WordPress.

Na stagingu zbierz zanonimizowane nagłówki dla dwóch kont o odmiennych ACL. Wymuś odpowiedzi 401, 403, 429 i 5xx oraz porównaj `cold`, `warm` i `bypass`. Status nie może zmienić prywatności odpowiedzi. Sprawdź `origin` i zewnętrzny brzeg infrastruktury. Nie wysyłaj cookie lub tokenu do Codexa; w raporcie wystarczy nazwa scenariusza, status i bezpieczne pola nagłówków.

N+1 jest hipotezą do sprawdzenia

Lista realizacji może generować koszt rosnący z liczbą rekordów. Zanim dodasz cache lub zmienisz SQL, sprawdź profil zapytań. Sama pętla z `get_post_meta` nie dowodzi N+1, bo WordPress może wcześniej zapełnić cache.

Fikcyjna obserwacja PERF-02 zawiera `posts_per_page=-1` i odczyty meta w pętli. To sygnał do pomiaru, nie wynik licznika zapytań. Zbierz liczbę i czas zapytań oraz powtarzające się wzorce. Porównaj listę z kilkoma i wieloma realizacjami przy zimnym i ciepłym cache. Jeśli wzrost wyniku z wielokrotnych wywołań dla tych samych rekordów, ustal, czy przyczyną jest wyłączony prefetch, autorski hook czy inny komponent.

Profiler działa na stagingu, bez publicznego debug display. Włączenie `SAVEQUERIES` w produkcji może zwiększać pamięć i ujawniać szczegóły, dlatego nie jest domyślną metodą utrzymania. Analizuj zapytania parametryzowane, indeksy i sposób sortowania. Nie proponuj dynamicznego SQL z interpolowanym wejściem użytkownika. Zmiana indeksu wymaga oceny migracji i danych; polecenie wygenerowane przez AI nie jest wystarczającą podstawą operacji na działającej bazie.

PROMPT DO CODEXA / Codex: diagnoza kosztu listy

Wklej do Codexa wraz z wymienionymi plikami. Oczekuj propozycji do przeglądu; wykonanie zmian wymaga własnego stagingu.

Przeczytaj implementację `wp-demo-projects` z mojego stagingu oraz zanonimizowany profil zapytań, który dostarczę. Jeśli profil nie jest załączony, jawnie wskaż brak dowodu. Znajdź miejsca nieograniczonego pobierania i hipotezy N+1, odróżniając działanie meta cache od rzeczywistych powtórzeń. Zaproponuj małą zmianę, plan pomiaru i test paginacji. Nie generuj migracji wykonującej się automatycznie.

KONTROLA: Odbiór: porównanie używa tego samego zbioru i stanu cache; spadek zapytań nie jest wymyślony. Zachowane są prawa dostępu i opublikowany status.

04 / OGRANICZANIE PRACY

Paginacja i object cache

Ograniczenie liczby pobieranych rekordów często jest czytelniejsze niż coraz większy cache. Paginacja ma jednocześnie poprawić koszt i zachować przewidywalny dostęp do wszystkich opublikowanych realizacji, także przy podobnych datach publikacji.

Wybierz rozmiar strony zgodny z interfejsem, na przykład dwanaście rekordów, oraz stabilne sortowanie. Liczba jest parametrem projektu, nie zmierzonym optimum. Jeśli potrzebujesz nawigacji z liczbą stron, zachowaj obliczenie całkowitej liczby wyników. `no_found_rows` pasuje do widoku, który naprawdę nie potrzebuje total; mechaniczne włączenie go wszędzie może zepsuć paginację. Użytkownik powinien dotrzeć do dalszych rekordów także bez skryptów przewijania nieskończonego.

Object cache przechowuje obiekty i wyniki w obrębie logiki aplikacji. Cache trwałe między zadaniami wymaga odpowiedniego backendu i konfiguracji. Dodanie Redis lub innej usługi nie naprawia złej reguły prywatności i nie usuwa potrzeby profilowania. Zapisz, jakie klucze wygasają po zmianie rekordu, kto odpowiada za backend oraz jak zachowuje się system przy jego niedostępności. Unikaj wspólnego klucza dla danych zależnych od uprawnień.

Odbiór obejmuje pierwszą, kolejną i ostatnią stronę, pusty wynik, usunięty rekord oraz zmianę daty publikacji. Porównaj identyfikatory, nie tylko wizualną liczbę kart. Nie czyść globalnie cache bez przyczyny na działającym serwisie; może to wywołać nagły wzrost pracy. Na stagingu wykonaj kontrolowany cold start, zapisz jego koszt i zestaw go ze zwykłym ruchem ciepłym. Raport powinien pokazać również koszt odnowienia cache.

- Publiczne listy odczytują wyłącznie opublikowane rekordy właściwego CPT.
- Wejściowy numer strony musi mieć granice i poprawną obsługę błędów.
- Prawa dostępu oraz aktualność wersji ważniejsze niż korzystny wynik benchmarku.

04 / INTERAKCJE

INP: mniej pracy po kliknięciu

Użytkownik czuje opóźnienie również wtedy, gdy strona już wygląda na gotową. Interakcja powinna szybko pokazać stan i umożliwić dalszą pracę. Nie myl oczekiwania na odpowiedź sieciową z czasem blokowania głównego wątku.

Przejdź kluczowe działania: otwarcie menu, filtr realizacji, formularz, wysłanie pytania do asystenta. Zarejestruj ślad interakcji na urządzeniu mobilnym lub porównywalnym profilu. Poszukaj długich zadań, kosztownego przeliczania stylów, dużego DOM i skryptów uruchamianych bez potrzeby. Duży pakiet analityczny potrafi blokować reakcję, nawet jeśli szybki cache dostarcza HTML. Raport powinien wskazać konkretną interakcję i przyczynę, a nie ogólne hasło optymalizacja JS.

Przy asystencie natychmiast pokaż potwierdzenie przyjęcia pytania i stan oczekiwania. Unikaj podwójnego wysyłania, zachowaj dostępny fokus, pozwól anulować widok oczekiwania. Anulowanie przeglądarki nie musi anulować zadania po stronie serwera; komunikat i protokół powinny to uwzględniać. Nie blokuj interfejsu aż do zakończenia modelu. Długie operacje dziel lub przenoś tam, gdzie nie zatrzymują obsługi klawiatury i kliknięć.

Zmierz po każdej zmianie ten sam scenariusz. Usunięcie animacji może poprawić ślad, ale nie powinno zniszczyć informacji o stanie. Redukcja skryptów wymaga sprawdzenia formularza, edytora i zgód użytkownika. Nie wyłączaj wszystkich wtyczek w produkcji, aby szukać winnego. Na stagingu izoluj podejrzany składnik pojedynczo z planem powrotu, a porównanie opisuj jako test danego scenariusza laboratoryjnego. Rzeczywiste INP wymaga osobnego pomiaru terenowego.

- Stany interfejsu: gotowy, przyjęto, trwa, zakończono, błąd oraz wynik niejednoznaczny.
- Test klawiaturą i czytnikiem: komunikat, fokus i możliwość dalszej pracy.
- Mniej skryptów jest hipotezą; odbiór potwierdza ślad i zachowaną funkcję.

Role, capabilities i najmniejsze prawa

Każda operacja powinna mieć wyraźnego wykonawcę i potrzebne prawo. Nazwa roli ułatwia administrację, ale właściwy warunek dostępu sprawdza capability.

W audycie testuj dozwolone i zabronione działania na konkretnych rekordach.

Wp-demo-intake przyjmuje szkic od integracji z wdl_ingest. Użytkownik integracyjny nie ma `publish_posts` ani praw administratora. Nie wystarczy zapewnić w opisie, że model zwykle tworzy draft. Endpoint i rola muszą wymuszać tę granicę. Fikcyjne SEC-03 pokazuje nadmiarowe `publish_posts`, więc plan naprawy obejmuje przegląd mapy praw, test na stagingu oraz odbiór ścieżki publikacji. Zmiana jest przygotowywana, nie wykonywana automatycznie przez generator.

Materiały mają odrębne `edit_wdl_materials`, `publish_wdl_materials` i `read_private_wdl_materials` oraz `map_meta_cap`. Edycja konkretnego rekordu wymaga prawa do tego rekordu, a nie tylko ogólnego zalogowania. Asystent używa `read_wdl_knowledge`; grupy dostępu wyznacza WordPress po stronie serwera. Nie przyjmuj `group` z przeglądarki jako wiarygodnej informacji. Uprawnienie w interfejsie nie zastępuje sprawdzenia w REST i prywatnej usłudze, która filtruje źródła przed rankingiem.

PROMPT DO CODEXA / Codex: macierz praw WP Demo Lab

Wklej do Codexa wraz z wymienionymi plikami. Oczekuj propozycji do przeglądu; wykonanie zmian wymaga własnego stagingu.

Na podstawie kontraktu rozpisz konta: anonim, redaktor, integracja intake, użytkownik knowledge i administrator. Dla każdego sprawdź odczyt publiczny, zapis szkicu, publikowanie, edycję materiału, odczyt prywatnego źródła oraz ask. Wskaż dokładne capabilities i miejsce kontroli. Przygotuj testy odmowy dla poprawnego nonce bez prawa i dla konta z prawem do innego rekordu. Nie zmieniaj ról na produkcji.

KONTROLA: Odbiór: integracja tworzy wyłącznie draft, a `group` z wejścia nie nadaje dostępu. Każda odmowa ma oczekiwany status i nie ujawnia treści źródła.

05 / UTRZYMANIE

Aktualizacje, zależności i sekrety

Bezpieczeństwo zależy od utrzymania całego stosu. Lista wtyczek jest początkiem audytu, nie jego końcem. Potrzebujesz wersji, właściciela, źródła dystrybucji, potrzeb biznesowych i sprawdzonej procedury aktualizacji dla każdego składnika.

Zbierz rdzeń, motyw, wtyczki, PHP, bibliotekę gateway i usługę Python. Określ, które komponenty są aktywne i potrzebne, a które pozostały po eksperymencie. Sam brak dostępnej aktualizacji nie dowodzi bezpieczeństwa. Własny przegląd powinien uwzględnić oficjalne informacje dostawców i zależności. Nie wpisuj fikcyjnego numeru podatności do raportu. Jeśli nie ma danych o wersji lub pochodzeniu, zapisz lukę inwentaryzacji i plan jej zamknięcia.

Aktualizację przygotuj po kopii i próbie odtworzenia: staging, zgodność, scenariusze funkcjonalne, kryteria przerwania i rollback. Dla pilnej poprawki bezpieczeństwa właściciel może przyjąć krótszą ścieżkę, lecz powinien udokumentować decyzję i ryzyko. Nie proponuj wyłączenia wszystkich wtyczek na działającym serwisie. Sprawdzanie sum rdzenia jest kontrolą integralności wybranych plików, nie pełnym skanerem autorskich modułów ani dowodem braku kompromitacji.

Klucze modelu, WDL_RAG_TOKEN, hasła bazy i token integracji przechowuj po stronie serwera w kontrolowanej konfiguracji lub menedżerze sekretów. Repo zawiera nazwę zmiennej i instrukcję, bez wartości. Nie drukuj sekretu podczas debugowania, nie kopiuj nagłówka Authorization do promptu. Po ujawnieniu potrzebne są rotacja i przegląd użycia, nie tylko usunięcie linii z ostatniego commita. Sprawdź także kopie, logi i artefakty wdrożenia.

- Manifest wersji i źródeł jest dowodem inwentaryzacji, nie certyfikatem bezpieczeństwa.
- Właściciel zatwierdza zmianę i czas wdrożenia; agent przygotowuje plan oraz różnice.
- Sekrety mają rotację, ograniczony dostęp i procedurę odtworzenia poza publicznym repo.

05 / GRANICE DOSTĘPU

REST, nonce i publiczne uploads

Nonce chroni określony kontekst żądania, ale nie nadaje praw. Publiczny adres pliku pozostaje publiczny niezależnie od stanu rekordu w panelu. Te dwie pomyłki potrafią zamienić poprawny interfejs w pozorną ochronę.

Przy cookie auth frontend przekazuje X-WP-Nonce. Endpoint nadal sprawdza zalogowanie i właściwą capability, a następnie waliduje wejście. Odczyt ask obejmuje filtr ACL przed rankingiem i przed wysłaniem kontekstu do modelu. Odmowa nie może ujawniać fragmentu dokumentu ani pełnej listy źródeł. Test powinien zawierać poprawny nonce konta bez prawa, konto z inną grupą, brak sesji i nieprawidłowe parametry. Sam ukryty przycisk nie jest kontrolą.

Biblioteka z tomu 04 przechowuje materiały przeznaczone do publicznego udostępnienia. Draft CPT chroni rekord, lecz plik w uploads może mieć bezpośredni URL. Prywatne PDF-y przechowuj poza publicznym webroot lub w magazynie z osobną kontrolą dostępu; link tymczasowy także wymaga zaprojektowanej polityki. Audit sprawdza bezpośredni adres z pustą sesją. Robots, noindex i trudna nazwa pliku nie zastępują autoryzacji ani blokady serwera.

Żądanie prywatne: kolejne warunki

01 Sesja

Tożsamość, właściwy kontekst uwierzytelnienia.

02 Capability

Prawo do operacji i konkretnego rekordu.

03 Walidacja i ACL

Dopuszczalne wejście; źródła dostępne przed rankingiem.

04 Odpowiedź

Minimalne dane i private/no-store także przy błędzie.

Własne zabezpieczenia uploadów obejmują rozmiar, typ, uprawnienie i sposób serwowania, nie tylko rozszerzenie nazwy. Nie zezwalaj na wykonanie kodu z katalogu uploadów. Zaproponuj konfigurację do oceny administratora i test na stagingu. W tej publikacji nie uruchomiono serwera ani walidatora PHP, dlatego odbiór tej granicy pozostaje zadaniem czytelnika.

Kopia to komplet spójnych danych

Posiadanie pliku archiwum nie oznacza gotowości do odzyskania serwisu. Kopia musi obejmować dane z tego samego punktu i wszystkie składniki potrzebne do uruchomienia strony, przepływów prywatnych oraz kont użytkowników.

WordPress wymaga bazy i plików: publicznych uploads, motywu, wtyczek i konfiguracji. WP Demo Lab dodaje prywatne źródła, indeks SQLite, wersje dokumentów, kolejkę oraz trwały stan idempotencji intake. Część indeksu można odbudować, ale trzeba znać wersję źródeł i modelu oraz przewidywany czas. Brak prywatnych plików przy odtworzonej bazie może dać stronę wyglądającą poprawnie, która nie potrafi odpowiedzieć na pytania uprawnionych użytkowników.

Ustal punkt spójności zgodnie z infrastrukturą: kontrolowane zatrzymanie zapisu lub spójny snapshot. Kopiowanie katalogu podczas aktualizacji bazy nie gwarantuje zgodnego zestawu. Opisz RPO jako dopuszczalną utratę danych i RTO jako czas odzyskania. To cele uzgodnione z właścicielem, dopóki nie wykonano próby. Nie podawaj fikcyjnego czasu restore. Manifest oraz suma kontrolna potwierdzają zawartość i niezmienność, lecz nie dowodzą poprawnego uruchomienia.

PROMPT DO CODEXA / Codex: sprawdź kompletność runbooka

Wklej do Codexa wraz z wymienionymi plikami. Oczekuj propozycji do przeglądu; wykonanie zmian wymaga własnego stagingu.

Przeczytaj backup-restore-runbook.md i mapę systemu. Zbuduj listę magazynów, zależności i kroków odtworzenia dla plików, DB, źródeł prywatnych, indeksu oraz idempotencji. Wskaż miejsca wymagające spójnego snapshotu, izolacji sieci i bezpiecznego odtworzenia sekretów. Nie twórz dumpu i nie uruchamiaj restore. Przygotuj protokół z pustymi polami czasu i wyniku, bez wymyślania RTO/RPO.

KONTROLA: Odbiór: wszystkie magazyny mają właściciela i test. Sekrety nie są drukowane, a hash nie jest opisany jako potwierdzenie działającej kopii.

06 / PRÓBA RESTORE

Odtworzenie w izolacji

Odtworzenie sprawdza użyteczność kopii i odsłania ukryte zależności. Środowisko próby musi być odseparowane, aby przywrócony serwis nie wysłał wiadomości, nie uruchomił zadań i nie zapisał danych do prawdziwego systemu.

Przygotuj staging z kontrolą dostępu i zablokowanymi wyjściami operacyjnymi. Wyłącz prawdziwe maile, webhooki, schedulery i provider AI; zastosuj kontrolowany mock. Noindex jest dodatkowym sygnałem dla wyszukiwarek, nie zabezpieczeniem kopii bazy. Odtwórz pliki i DB ze zgodnego punktu, dostosuj konfigurację bez ujawniania haseł. Zmianę adresów wykonuj narzędziem świadomym serializacji, po dry-run i kopii, a nie globalną podmianą SQL w dowolnym tekście.

Zaloguj się, sprawdź pięć stron, obrazy i menu. Następnie przejdź prawa domenowe: realizacje publiczne, materiały i wersje, intake draft, ograniczenia integracji, RAG dla dwóch grup oraz odmowy. Usuń testowy dokument i potwierdź, że stare fragmenty nie wracają. Nie ograniczaj odbioru do statusu 200 na HOME. Odtworzona strona może wyglądać dobrze mimo uszkodzonego indeksu, kolejki lub źródeł przechowywanych poza WordPress.

Próba odtworzenia

01 Izolacja

Staging, auth i brak wysyłek produkcyjnych.

02 Spójna kopia

Pliki, baza oraz prywatne magazyny.

03 Kontrole funkcji

Role, szkice, materiały i ACL asystenta.

04 Protokół

Zmierzony czas, braki i decyzja właściciela.

Stan unknown w kolejce lub intake wymaga uzgodnienia, zanim uruchomisz powtórzenie. Nie zakładaj exactly-once przy nieznanym wyniku i efektach hooks. Protokół zapisuje faktyczny czas, zakres utraty danych oraz nierozwiązane przypadki. W pakiecie sprawdzono kompletność instrukcji, ale nie wykonano prawdziwego odtworzenia; wszystkie pola realizacji pozostają do wypełnienia na własnym stagingu.

06 / ZMIANA KONTROLOWANA

Staging, wdrożenie i rollback

Naprawa jest gotowa do wdrożenia dopiero wtedy, gdy wiadomo, jak sprawdzić efekt i jak wrócić do działającego stanu. Cofnięcie plików nie zawsze cofa zmianę danych, schematu lub zewnętrznego indeksu.

Przygotuj niewielki pakiet zmian i manifest wersji. Odnotuj zależności: poprawa cache może wymagać reguł CDN; aktualizacja wtyczki może mieć migrację bazy; indeks zależy od modelu i wersji dokumentów. Zdefiniuj próg przerwania, osobę dyżurną oraz okno monitoringu. Prosty plan powrotu ma wskazać poprzedni pakiet, sposób przełączenia ruchu i postępowanie z zapisami powstałymi po wdrożeniu. Nie obiecuj cofnięcia wszystkiego jednym przyciskiem.

Staging powinien odwzorować ważne warstwy produkcji, lecz używać bezpiecznych danych i odizolowanych integracji. Sam lokalny serwer bez CDN nie sprawdzi błędnej reguły na brzegu sieci. Dla każdego testu zapisz, które warstwy były obecne. Porównanie laboratoryjne before/after ma używać tych samych warunków; odbiór prywatności wymaga dwóch sesji. Rzeczywiste CWW obserwuj po wdrożeniu przez właściwy okres, zamiast obiecywać natychmiastowy terenowy rezultat ze stagingu.

PROMPT DO CODEXA / Codex: plan wdrożenia bez automatycznych napraw

Wklej do Codexa wraz z wymienionymi plikami. Oczekuj propozycji do przeglądu; wykonanie zmian wymaga własnego stagingu.

Dla ustaleń SEC-01, SEC-02 i PERF-01 przygotuj plan zmian na stagingu. Wymień pliki lub reguły do przeglądu, zależności, backup, testy pozytywne i negatywne, warunki przerwania i rollback. Rozdziel cofnięcie kodu, konfiguracji oraz danych. Nie aktualizuj, nie usuwaj i nie publikuj. Wynik ma być pakietem decyzji do zatwierdzenia przez właściciela.

KONTROLA: Odbiór: każda zmiana ma dowód i drogę powrotu. Prawdziwe nowe zapisy oraz migracje DB nie są pominięte, a test mocka nie zastępuje WordPress.

Adresy, canonical i statusy

Techniczne SEO zaczyna się od spójnych adresów i dostępnej treści. Canonical wskazuje preferowany wariant podobnych stron, ale nie jest mechanizmem autoryzacji ani gwarancją, że wyszukiwarka wybierze dokładnie wskazany URL.

Dla każdej publicznej strony ustal właściwy adres, HTTPS, status odpowiedzi i źródło canonical. Na Usługach przykładowy błąd SEO-01 wskazuje staging.example.test. Plan naprawy obejmuje konfigurację domeny i jednego odpowiedzialnego generatora meta, ponieważ motyw i wtyczka SEO mogą wytworzyć dwa sprzeczne znaczniki. Sprawdź finalny HTML, ewentualny nagłówek Link, mapę witryny i linki wewnętrzne. Sama edycja pola w panelu nie dowodzi prawidłowego wyniku.

Adres docelowy powinien odpowiadać treści i działać jako publiczny właściwy wariant. Nie kieruj wszystkich paginowanych realizacji na pierwszą stronę bez oceny ich treści i funkcji. Dla rzeczywistej zmiany adresu rozważ przekierowanie zgodne z celem migracji, zamiast pozostawiać łańcuch wielu skoków. Nie zamieniaj brakującej strony w soft 404 z komunikatem błędu i statusem 200. Usunięty zasób i tymczasowa awaria to inne sytuacje.

PROMPT DO CODEXA / Codex: kontrola sygnałów URL

Wklej do Codexa wraz z wymienionymi plikami. Oczekuj propozycji do przeglądu; wykonanie zmian wymaga własnego stagingu.

Dostarczę zanonimizowany HTML pięciu stron i sitemap ze stagingu. Sprawdź statusy, canonical, linki menu, adresy mediów i domenę. Jeśli nie dostarczę statusu HTTP, nie zgaduj go z HTML. Wskaż konflikty motywu i SEO oraz adresy stagingu. Przygotuj propozycję poprawy i macierz odbioru bez wysyłania żądań. Nie obiecuj indeksacji ani wzrostu pozycji.

KONTROLA: Odbiór: publiczne adresy są spójne, canonical dotyczy treści danego widoku, a niezbadane statusy pozostają nieustalone.

07 / WIDOCZNOŚĆ

Robots, noindex i ochrona stagingu

Kontrola indeksowania dotyczy robotów wyszukiwarek, a ochrona dostępu dotyczy osób i programów. Staging z danymi użytkowników wymaga uwierzytelnienia lub blokady infrastruktury. Sam zakaz indeksowania nie ukrywa pliku przed znajdującym adres.

Robots.txt instruuje o crawlowaniu; noindex w meta robots lub nagłówku X-Robots-Tag instruuje o indeksowaniu. Jeśli robot nie może pobrać strony przez blokadę robots.txt, może nie zobaczyć noindex. Nie buduj więc planu usunięcia z indeksu na sprzecznych warunkach. W realnym środowisku dobrać metodę do sytuacji i sprawdzić ją narzędziami wyszukiwarki. W demonstracji nie istnieje działająca domena ani raport indeksowania, zatem nie deklarujemy wyniku.

Staging chronimy oddzielną warstwą auth lub ograniczeniem sieci; noindex jest dodatkową kontrolą. Publiczna produkcja powinna mieć zamierzony stan indeksowania po wdrożeniu. Sprawdź ustawienie widoczności WordPress, plugin SEO, nagłówki serwera i reguły CDN. Częsty błąd polega na przeniesieniu noindex ze stagingu na produkcję albo pozostawieniu stagingowego canonical. Protokół odbioru musi zawierać oba środowiska i dokładne miejsca konfiguracji, nie tylko jeden checkbox w panelu.

Dla publicznego PDF możesz użyć odpowiedniego nagłówka indeksowania, ale prywatny PDF musi odmówić dostępu bez uprawnienia. Nie mieszaj tych celów w jednej poradzie. Jeśli tajny plik był publiczny, potrzebne są usunięcie ekspozycji, przegląd dostępu i odpowiednia procedura danych; zmiana noindex nie cofa wcześniejszego ujawnienia. AI może przygotować checklistę i rozpoznać konflikt, lecz operator potwierdza rzeczywiste odpowiedzi serwera.

- Test stagingu: pusta sesja nie odczytuje HTML ani danych kopii.
- Test produkcji: zamierzone indeksowanie i brak domeny stagingu.
- Test PDF: osobno autoryzacja bezpośredniego URL i polityka indeksowania.

Sitemap i linki wewnętrzne

Mapa witryny pomaga odkrywać adresy, lecz nie zastępuje nawigacji i nie gwarantuje indeksowania. Dla małej firmy sprawdź najpierw, czy ważne strony są dostępne z czytelnych linków i mają użyteczną, aktualną treść.

HOME prowadzi do Usług i Realizacji, a treść usług prowadzi do właściwego kontaktu. O firmie opisuje odpowiedzialność i sposób pracy. Linki powinny mieć sensowny tekst, dostępny także bez odgadywania z ikonki. Publiczna biblioteka może łączyć materiał z tematem i wersją. Unikaj tworzenia wielu podobnych stron wyłącznie pod frazy, jeżeli nie wnoszą odrębnej informacji. Audyt porównuje mapę treści z rzeczywistymi potrzebami użytkownika, nie tylko liczbę URL-i.

Sprawdź działającą mapę XML, publikowane typy rekordów i adresy canonical. Do mapy nie dodawaj szkiców intake, prywatnych źródeł ani testowych kopii. Jeżeli używasz wtyczki SEO, ustal, czy zastępuje mapę rdzenia, aby nie analizować nieaktualnego endpointu. Daty modyfikacji mają odpowiadać istotnym zmianom. Masowa aktualizacja lastmod bez zmiany treści nie daje użytkownikowi nowej wartości. Generator raportu nie sprawdza mapy online, więc potrzebuje dostarczonego eksportu.

Przejdź strony z menu, linków treści i paginacji; zapisz miejsca osierocone oraz pętle. Sprawdź statusy linków i dostępność zasobów potrzebnych do renderowania. Wyszukiwarka powinna widzieć zamierzoną publiczną treść, a konta prywatne nadal muszą mieć kontrolę dostępu. Raport może połączyć mapę adresów z właścicielami treści i terminem przeglądu. To ułatwia utrzymanie po wdrożeniu oraz pozwala znaleźć przyczynę braku aktualizacji, zamiast automatycznie mnożyć strony.

- Mapa: tylko przeznaczone do publikacji, właściwe adresy działających stron.
- Linki: opisowy tekst, rozsądna liczba kroków i działająca paginacja.
- Odbiór: treść, indeksowanie i autoryzacja sprawdzane jako osobne warunki.

Organization i Service opisują fakty

Dane strukturalne są czytelnym opisem istniejącej treści. WP Demo Lab może opisać organizację i usługę projektowania stron, ale nie powinien dopisywać ocen, certyfikatów, adresów czy obietnic, których nie ma w demonstracyjnym briefie.

Przykład service-organization.jsonld zawiera Organization z nazwą i URL oraz Service z nazwą, URL i provider wskazującym identyfikator organizacji. To demonstracyjne dane fikcyjnej firmy. Dla prawdziwej witryny wartości pochodzą z zatwierdzonego źródła i odpowiadają widocznej treści. Stabilne @id pozwala wiązać encje bez powielania różnych opisów tej samej organizacji. Nie wkładaj do markupu prywatnego numeru telefonu redaktora ani niepotwierdzonej lokalizacji.

KOD W PRAKTYCE / JSON / Istotny fragment grafu Service

```
{
  "@context": "https://schema.org",
  "@type": "Service",
  "name": "Projektowanie stron WordPress",
  "url": "https://example.test/uslugi/",
  "provider": {
    "@id": "https://example.test/#organization"
  }
}
```

PLIK: service-organization.jsonld

Fragment pełnego grafu demonstracyjnego; nie wynik testu wyszukiwarki.

Poprawny JSON nie oznacza poprawnego znaczenia ani gwarantowanego rozszerzonego wyniku. Nie każdy typ schema ma osobną funkcję prezentacji w Google; samo Service nie obiecuje rich result. Narzędzia walidacyjne sprawdzają różne poziomy, dlatego odbiór łączy składnię, zgodność faktów i wymagania konkretnej funkcji. Unikaj wymyślnego aggregateRating z SEO-02. Lepszy jest krótki prawdziwy opis niż rozbudowany graf zawierający marketingową fikcję.

08 / JAKOŚĆ

Walidacja treści i źródła prawdy

AI sprawnie wypełnia strukturę, ale może wypełnić również brakujące fakty. Przy SEO brak informacji musi pozostać brakiem do wyjaśnienia. Redaktor powinien zatwierdzić dane firmy i usług zanim zostaną opublikowane w widocznym tekście lub JSON-LD.

Utwórz tabelę wartości i źródeł: nazwa organizacji z briefu, adres publiczny z konfiguracji, nazwa usługi z zatwierdzonej strony, dostępność oferty od właściciela. Dla pól niepotwierdzonych wpisz brak danych zamiast wygenerowanej odpowiedzi. Dotyczy to szczególnie ocen klientów, nagród, cen, godzin i zasięgu działania. Wersja demonstracyjna nie zawiera tych wartości, więc agent nie ma podstaw, aby je wymyślać. Źródło prawdy powinno mieć właściciela i datę przeglądu.

Przejrzyj, czy wtyczka SEO, motyw i autorski moduł nie generują równocześnie rozbieżnych opisów. Nie usuwaj całego systemu SEO bez oceny zależności. Zaproponuj jednego właściciela grafu lub spójny sposób rozszerzania. Sprawdź końcowy HTML i treść strony, nie wyłącznie plik JSON na dysku. Walidacja offline tego pliku potwierdza strukturę przykładu; nie potwierdza, że serwer rzeczywiście go emituje w poprawnym miejscu.

PROMPT DO CODEXA / Codex: JSON-LD bez dopisywania faktów

Wklej do Codexa wraz z wymienionymi plikami. Oczekuj propozycji do przeglądu; wykonanie zmian wymaga własnego stagingu.

Przeczytaj service-organization.jsonld i zatwierdzony opis usług. Dla każdej wartości wskaż źródło. Usuń z propozycji pola bez dowodu, zwłaszcza oceny, adres i certyfikaty. Sprawdź spójność @id, provider i URL oraz potencjalne duplikaty generatorów. Zwróć poprawny JSON-LD i listę pytań do redaktora, nie publikuj. Oddziel poprawność składni od uprawnienia do rich result.

KONTROLA: Odbiór: graf zawiera wyłącznie istniejące fakty; treść widoczna i markup mówią to samo. Walidacja nie jest przedstawiona jako gwarancja pozycji w wyszukiwarce.

Treści AI przechodzą przez draft

Automatyczne przygotowanie szkicu skraca pracę, gdy pozostawia kontrolę publikacji człowiekowi. Audyt SEO obejmuje ten przepływ, bo szybko produkowana treść może powielić adresy, wymyślić informacje i rozmyć jakość całej witryny.

Formularz zbiera uzgodnione wejście, walidator normalizuje dane, a adapter AI tworzy propozycję. Intake przyjmuje request_id, title, content i stałe source=demo-form, wymaga wdl_ingest oraz zapisuje draft. Redaktor sprawdza sens, fakty, prawa do materiału, linki, tytuł i miejsce w architekturze treści. AI nie otrzymuje prawa samodzielnej publikacji. Ta sama zasada dotyczy zmian canonical i danych strukturalnych: są częścią odbioru, a nie niekontrolowanym skutkiem generowania.

Trwała idempotencja chroni powtarzalny request_id, lecz nie obiecuje exactly-once przy nieznanym wyniku i skutkach hooks. Timeout wymaga sprawdzenia stanu oraz uzgodnienia, czy szkic już powstał. Ślepe retry może stworzyć duplikaty, które później trafią do mapy witryny lub pracy redaktora. Test odbioru sprawdza rekordy i statusy, nie tylko komunikat formularza. Nazwa zadania i identyfikator mają wystarczyć do diagnozy bez kopiowania całej prywatnej treści do logów.

Dla nowych stron ustal odbiorcę, pytanie użytkownika i odpowiedź opartą na wiedzy firmy. Nie twórz serii bliskich kopii wyłącznie po to, aby zwiększyć liczbę słów kluczowych. Sprawdź czytelność, spójność z ofertą i datę ważności. Raport może wskazać brak procesu redakcyjnego równie jasno jak błędny nagłówek HTTP. Właściciel treści zatwierdza publikację i przyszły przegląd; monitoring techniczny nie zastąpi tego obowiązku.

- Szkic nie trafia do sitemap ani publicznego indeksu materiałów.
- Nieznany wynik intake ma ścieżkę uzgodnienia bez automatycznej publikacji.
- Odbiór obejmuje jakość treści, linki, fakty i dopiero potem ustawienia SEO.

Monitoring zaczyna się od pytań

Monitoring ma odpowiedzieć, czy użytkownik może wykonać ważne zadanie i czy zmiana pogorszyła system. Jedna zielona kontrolka serwera nie dowodzi działania formularza, indeksu wiedzy, praw dostępu ani aktualności materiałów.

Rozdziel dostępność publicznej strony, poprawność funkcji i zdrowie prywatnych usług. Sprawdzenie HOME może potwierdzić HTTP, lecz asystent wymaga gateway, indeksu, ACL i provider. Publiczny status nie ujawnia konfiguracji, tokenów, nazw prywatnych źródeł ani szczegółów kont. Kontrola wewnętrzna może podawać więcej diagnostyki, jeśli ma osobną autoryzację. Ustal, kto odbiera alert i jaki ma pierwszy krok; samo zbieranie wykresów nie tworzy procedury reakcji.

Przed wdrożeniem zapisz baseline: błędy, czas odpowiedzi, długość kolejki i podstawowe profile lab. p95 czasu prywatnej usługi jest przydatną metryką operacyjną, ale nie zastępuje p75 CWV. Obie dotyczą innych zdarzeń i różnych pytań. Dla alertu wybierz okno, minimalną liczbę zdarzeń i warunek powrotu, aby ograniczyć szum. Próg musi wynikać z własnej funkcji i ruchu, nie z fikcyjnego benchmarku książki.

PROMPT DO CODEXA / Codex: plan monitoringu po zmianie

Wklej do Codexa wraz z wymienionymi plikami. Oczekuj propozycji do przeglądu; wykonanie zmian wymaga własnego stagingu.

Dla architektury WP Demo Lab przygotuj sygnały dostępności, poprawności i kosztu. Rozdziel publiczne strony, intake, ask, indeks i worker. Zaproponuj bezpieczne pola logu, odbiorcę alertu, pierwszy krok diagnostyczny i warunek rollback. Nie twórz działającego endpointu ani progów opisanych jako zmierzone. Wyraźnie odróżnij p95 usługi od p75 Core Web Vitals.

KONTROLA: Odbiór: alert prowadzi do działania, publiczny status nie ujawnia sekretów, a metryka ma zakres, okno oraz właściciela.

09 / DANE I RETENCJA

Logi bez prywatnych pytań

Log techniczny powinien wystarczyć do diagnozy bez kopiowania całej rozmowy. Pytania, dokumenty, cookies i tokeny mogą ujawnić więcej niż pierwotny błąd. Najbezpieczniej ograniczyć dane już przy zapisie, a potem kontrolować ich dostęp i czas życia.

Fikcyjne OPS-01 zapisuje question i pełną response. Plan naprawy usuwa treść z formatu, pozostawiając identyfikator żądania, czas, typ operacji, status, kod błędu, wersję adaptera i techniczny licznik kosztu. Jeżeli korelacja wymaga identyfikatora użytkownika, oceń niezbędność i sposób pseudonimizacji. Nawet pseudonim może pozostać daną osobową, więc nie traktuj go jako automatycznego zwolnienia z polityki. Unikaj także pełnego URL z prywatnymi parametrami.

Ograniczenie nowego zapisu nie usuwa starych kopii. Ustal retencję, rolę odczytu, kopie i eksporty. Zanonimizowany przykład przekazany do AI powinien zawierać tylko pola potrzebne do diagnozy; nie maskuj jednego tokenu, zostawiając drugi w treści błędu. Raport wskazuje miejsca, właściciela i bezpieczną procedurę przeglądu. Generator nie wykonuje usuwania danych, bo nie zna wymogów utrzymania ani sytuacji incydentu. Zmiany wymagają świadomej decyzji operatora.

Usunięcie materiału obejmuje rekord, publiczny plik zgodnie z decyzją publikacyjną, prywatne źródło, fragmenty indeksu i zależne magazyny. Kopie zapasowe mają własną politykę wygaśnięcia oraz sposób zapobiegania ponownemu pojawieniu się usuniętych danych po restore. Nie obiecuj natychmiastowej eliminacji wszystkich kopii jednym poleceniem. Przetestuj odczyt po usunięciu i restart usługi, zapisując tylko identyfikatory. Logi odbioru również powinny przestrzegać minimalizacji.

- Zapisuj status i identyfikator, nie sekret, cookie ani prywatny kontekst RAG.
- Rozróżnij retencję produkcji, stagingu, logów, eksportów i backupu.
- Po zmianie sprawdź faktyczną próbkę logów; dokumentacja formatu nie jest dowodem wykonania.

09 / BUDŻET

Koszt AI, kolejka i wynik nieznany

Koszt asystenta zależy od liczby żądań, kontekstu, modelu i ponowień. Szybki frontend nie powinien uruchamiać nieograniczonej pracy serwera. Limity są częścią architektury, a nie dodatkiem po pierwszym wysokim rachunku.

Ustal budżet i jednostkę rozliczenia: żądanie, użytkownik, dzień lub projekt. Nie podajemy aktualnej ceny modelu, bo przykład offline nie korzysta z providera. Kalkulacja własnego systemu wymaga sprawdzonego cennika i pomiaru liczby tokenów. Zastosuj granicę kontekstu, maksymalną liczbę fragmentów, czas oraz równoległość workerów. Długi kontekst może zwiększyć koszt, nawet gdy odpowiedź dla użytkownika jest krótka. Rate limit musi działać po stronie serwera.

Kolejka ma czytelne stany: pending, processing, completed, failed i unknown. Ostatni oznacza niejednoznaczny efekt, a nie automatyczne zaproszenie do retry. Intake wymaga uzgodnienia trwałego request_id przed ponowieniem. Dla zapytań odczytowych również oceń ryzyko podwójnego kosztu i granice ponowień. Backend powinien odrzucać nadmiar zgodnie z polityką i zwracać bezpieczny komunikat; frontend pokazuje oczekiwanie bez wielokrotnego wysyłania. Anulowanie widoku nie zawsze zatrzyma zadanie.

Kontrolowana praca AI

01 Wejście

Walidacja, prawo, limit użytkownika.

02 Kolejka

Trwałe ID i ograniczona równoległość.

03 Worker

Timeout, budżet i zapis stanu.

04 Uzgodnienie

Unknown analizowany przed retry.

05 Odbiór

Wynik lub szkic zatwierdzany przez człowieka.

Ćwiczenie może obciążać deterministyczny mock i sprawdzić sterowanie kolejką, lecz nie zmierzy opóźnień prawdziwego modelu ani jakości embeddings. Wyniki przedstawiaj oddzielnie. Kosztowy cel po naprawie pozostaje planem, dopóki własny pomiar nie potwierdzi zmniejszenia pracy przy zachowaniu jakości, ACL i przewidywalnego zachowania błędów.

Pełny raport z generatora

Uruchomienie generatora porządkuje dziesięć demonstracyjnych ustaleń według ryzyka, dopisuje cele i warunki odbioru. Wynik powinien być czytelny dla właściciela strony i technika, a brak pomiaru pozostać widoczny w obu formatach.

Wejście zawiera `schema_version=1`, `site`, `evidence_mode`, `cwv` i `findings`. Każde ustalenie ma ID, kategorię, dowód, stan przed, cel, działanie, test i właściciela oraz dwie liczby ryzyka. Wynik jest deterministyczny: te same dane dają tę samą kolejność, najpierw wyższy score, potem ID. Nie ma losowych wartości ani daty generacji zmieniającej porównanie. CLI nie łączy się z serwerem; jedynie waliduje i zapisuje raport.

KOD W PRAKTYCE / BASH / Raport Markdown i JSON

```
python audit_report.py audit-fixture.json
python audit_report.py audit-fixture.json \
  --output report.md
python audit_report.py audit-fixture.json \
  --format json --output report.json
python -m unittest discover -s tests -v

PLIK: audit_report.py
Polecenia Python offline wykonano w pakiecie; nie uruchamiają WordPress.
```

Expected-report.md jest punktem porównania dla testu regresji, nie rzeczywistym before/after witryny. P1 obejmuje SEC-01, SEC-02 i OPS-01; P2 obejmuje pięć kolejnych ustaleń, a P3 dwa. Progi CWV są testowane na granicach, jednak fixture pozostaje unmeasured. Błędne liczby, powtórzone ID i niepełny kontekst pomiaru wywołują błąd. Własne dane wprowadź do osobnej karty, zachowując oryginalne eksporty.

10 / RAPORT DEMONSTRACYJNY

Plan napraw i porównanie przed/po

W pełnym audycie plan wynika z zależności, nie tylko z rankingu. Najpierw usuń ekspozycję danych i zapewnij odtworzenie, potem poprawiaj wydajność oraz sygnały SEO. W demonstracji wszystkie naprawy są celami planned.

Pierwsza partia obejmuje SEC-01: przegląd cache, wyjątek dla ask/intake oraz test z dwiema sesjami; SEC-02: spójną kopię i próbę restore; OPS-01: minimalny format logów i plan dla historii. Właściciele otrzymują konkretne warunki odbioru. Nadmiarowe prawa integracji oraz canonical wymagają równoległego przeglądu właściwych modułów, lecz wdrożenia powinny mieć kontrolowane granice. Dla realnej ekspozycji danych zastosuj własną procedurę incydentu, nie czekaj na estetykę raportu.

Obszar	Przed — fikcyjne	Po — cel, nie wynik
Ask	public cache	private/no-store i brak HIT
Restore	brak protokołu	udokumentowana próba w izolacji
Hero	lazy, jeden wariant	warianty i wymiary; pomiar
Canonical	adres stagingu	spójny URL publiczny
CWV	niezmierzone	pomiar field do zebrania

Druga partia porządkuje hero, okładki i paginację; trzecia dane strukturalne oraz kolejkę AI. Zmiany rozdziel, aby umieć przypisać regresję do konkretnej decyzji. W kolumnie po nie wpisuj hipotetycznych milisekund ani wzrostu pozycji. Własny zmierzony efekt powinien wskazać źródło, okres, profil oraz różnice zakresu. Jeżeli test nie został wykonany, pozostań przy planned i zapisz termin następnego kroku.

Checklistę raportu zamykają: zakres i wersje, architektura, dowody, priorytety z uzasadnieniem, właściciele, działania, rollback, wyniki oraz ograniczenia. Wyeksportuj JSON dla dalszej pracy i Markdown do przeglądu. Nie usuwaj ustalenia tylko dlatego, że poprawa jest trudna. Można zaakceptować ryzyko z podpisem właściciela i datą przeglądu; nie można opisać braku wykonania jako ukończonej naprawy.

10 / ZAMKNIĘCIE

Odbiór, testy i dalsza praca

Projekt kończy się kompletną decyzją: co sprawdzono, co pozostaje niepotwierdzone i kto podejmuje następny krok. Testy lokalnego narzędzia chronią raport, ale odbiór WordPress wymaga własnego środowiska i dowodów działania całej architektury.

W tym pakiecie wykonano unittest dla generatora, CLI, progów CWV, walidacji liczb, porządkowania ryzyk, oddzielenia illustrative/measured oraz kart prywatnych nagłówków. Sprawdzono również JSON-LD i strukturę plików. Nie wykonano PHP, aktywacji WordPress, PageSpeed, skanu bezpieczeństwa, prawdziwego CDN ani próby odtworzenia bazy. TEST_STATUS.md wymienia rzeczywiste wyniki i te ograniczenia. Czytelnik powinien dodać własny protokół, zamiast zmieniać opis testu offline w deklarację integracji.

Odbiór stagingu przechodzi ważne ścieżki publiczne, role, odmowy REST, draft intake, ACL przed rankingiem, aktualną wersję materiału, bezpośrednio URL plików i cache błędów. Pomiar wydajności opisuje profile oraz źródło. SEO wymaga spójności URL, zamierzonego indeksowania, mapy i prawdziwego JSON-LD. Kopia jest zaakceptowana dopiero po próbie restore. Monitoring i budżet AI mają właściciela oraz procedurę reakcji. Dopiero taki zestaw pozwala sensownie zatwierdzić wdrożenie.

PROMPT DO CODEXA / Codex: raport odbioru bez nieuprawnionych sukcesów

Wklej do Codexa wraz z wymienionymi plikami. Oczekuj propozycji do przeglądu; wykonanie zmian wymaga własnego stagingu.

Przeczytaj raport, TEST_STATUS.md i staging-checks.md. Zbuduj końcową listę: wykonano, nie wykonano, wynik niejednoznaczny, ryzyko zaakceptowane. Dla każdej pozycji wymagaj istniejącego dowodu i właściciela. Wykryj miejsca, gdzie plan nazwano rezultatem albo mock nazwano WordPress. Przygotuj krótki raport do akceptacji oraz plan następnych pomiarów. Nie wdrażaj i nie publikuj.

KONTROLA: Odbiór: każde stwierdzenie sukcesu ma dowód; prywatność, restore i rollback mają własne kryteria. Brak wyniku jest jawny, a dalsza praca ma konkretnego odpowiedzialnego.

BIBLIOGRAFIA / DOSTĘP 13.10.2026

Źródła i granice weryfikacji

Wybrane źródła prowadzą do oficjalnej dokumentacji. Zweryfikowano ich dostępność i kluczowe zasady użyte w tomie; numery wersji oraz konfigurację własnej witryny należy sprawdzić ponownie w dniu wykonania audytu.

Web Vitals

<https://web.dev/articles/vitals>

Progi LCP, INP i CLS oraz ocena 75. percentyla; rozdział 2.

Search Engine Optimization (SEO) Starter Guide

<https://developers.google.com/search/docs/fundamentals/seo-starter-guide>

Odkrywanie stron, canonical, linki i mapa witryny; rozdział 7.

General Structured Data Guidelines

<https://developers.google.com/search/docs/appearance/structured-data/sd-policies>

Zgodność danych strukturalnych z treścią i granice obietnic; rozdział 8.

Organization Schema Markup

<https://developers.google.com/search/docs/appearance/structured-data/organization>

Opis organizacji, identyfikacja i pola publiczne; rozdział 8.

Hardening WordPress

<https://developer.wordpress.org/advanced-administration/security/hardening/>

Oficjalne zasady zabezpieczenia WordPress; rozdział 5.

Backups

<https://developer.wordpress.org/advanced-administration/security/backup/>

Zakres kopii bazy i plików oraz odtwarzanie; rozdział 6.

Upgrading WordPress

<https://developer.wordpress.org/advanced-administration/upgrade/upgrading/>

Proces aktualizacji i przygotowanie kopii; rozdziały 5–6.

Dokumentacja określa mechanizmy i wymagania, ale nie potwierdza konfiguracji WP Demo Lab.

Wszystkie obserwacje fixture są fikcyjne. Wyniki testów Python dotyczą generatora i interpretacji kart wejściowych. Nie zastępują narzędzi wyszukiwarki, pomiarów terenowych ani przeglądu prawdziwych plików serwera. Service w przykładzie jest opisem demonstracyjnej usługi, nie obietnicą specjalnego wyglądu wyniku Google.

Dalsza praca: zachowaj raport demonstracyjny jako ćwiczenie, utwórz własną kartę manual, zbierz źródła i potwierdź je na stagingu. Nie wysyłaj prywatnych eksportów ani sekretów do promptu. Dla każdego środowiska zapisz datę, zakres oraz dokładne wersje. Gdy dokumentacja zmieni API lub wymagania, zaktualizuj testy odbioru i ponownie sprawdź ograniczenia. Stan bez pomiaru pozostaje poprawnym wynikiem rzetelnego audytu.