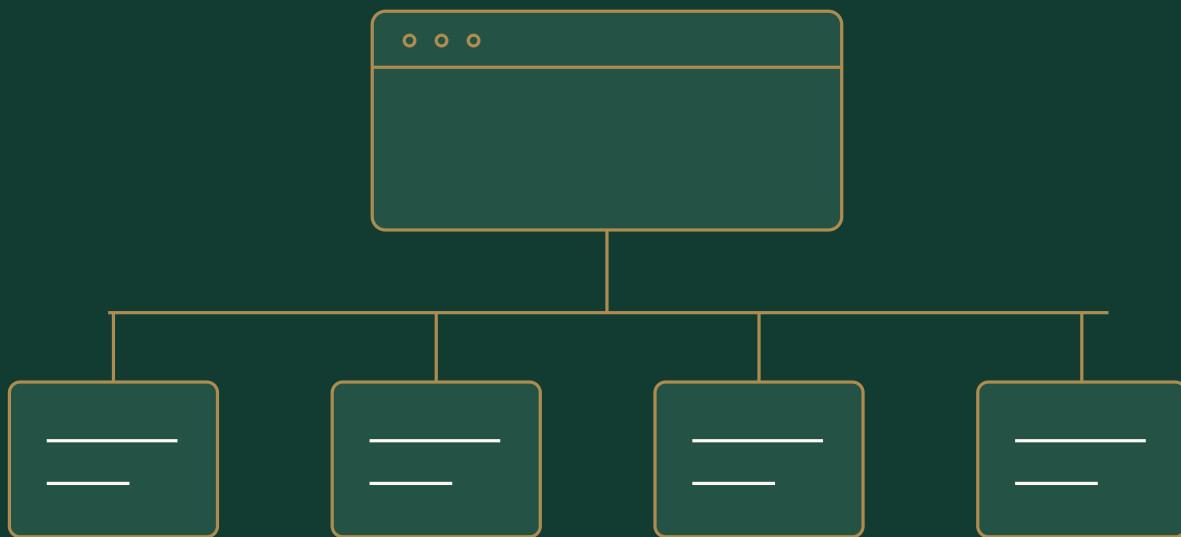


WORDPRESS + AI

Od pomysłu do profesjonalnej strony



PROJEKT TWORZY CZŁOWIEK.
AI WSPIERA PROCES.

01

10 ROZDZIAŁÓW / JEDEN SYSTEM DEMONSTRACYJNY

Spis treści

Wprowadzenie i mapa projektu	3
01 Brief, odbiorca i architektura informacji	4
Cel, który można sprawdzić · Pięć adresów, pięć odpowiedzi · Kryteria i odpowiedzialność	
02 WordPress: dane, motyw i odpowiedzialności	7
Co jest plikiem, a co rekordem · Motyw ma rozpoznawalną strukturę · Szablon oddziela ramę od treści	
03 Środowisko oraz powtarzalny początek pracy	10
Uruchom lokalne laboratorium · Utwórz treść, zanim ocenisz wygląd · Codex porządkuje diagnozę środowiska	
04 System wizualny zapisany w motywie	13
Paleta i typografia mają jedno źródło · Interfejs prowadzi do decyzji · CSS i edytor korzystają z tej samej paczki	
05 Gutenberg, komponenty i wzorce	16
Blok, wzorec i część szablonu · Hero jest wzorcem z nagłówkiem pliku · Zmiana komponentu ma mały zakres	
06 Pięć stron oraz redakcja z AI	19
Treść Home, O firmie i Usług · AI redaguje wyłącznie otrzymane fakty · Realizacje i Kontakt budują zaufanie	
07 Media, rytm i responsywny układ	22
Media wymagają planu przed importem · Układ zmienia się razem z przestrzenią · Obserwuj stronę z rzeczywistą treścią	
08 Dostępność w codziennej pracy	25
Focus musi być widoczny i użyteczny · Kontrast, nagłówki i kontekst · Zamień checklistę w scenariusze	
09 Staging, synchronizacja i wdrożenie	28
Staging jest miejscem uzgadniania zmian · Statyczna kontrola ma jasne granice · Synchronizacja plików i Site Editor	
10 Odbiór, dokumentacja i następny etap	31
Odbiór wymaga demonstracji właściciela · Po wdrożeniu sprawdź tę samą ścieżkę · Dowody i ograniczenia pierwszego tomu	
Źródła techniczne i dalsza lektura	34

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

JAK PRACOWAĆ Z TOMEM

Profesjonalna strona zaczyna się od decyzji

Zbudujesz jedną spójną witrynę fikcyjnej firmy. AI będzie wspierać planowanie, redakcję i przegląd kodu, a o publikacji zdecyduje człowiek.

WP DEMO LAB projektuje strony i przepływy treści dla małych firm. To celowo niewielki przykład: Home, O firmie, Usługi, Realizacje i Kontakt. Dzięki temu można prześledzić każdą decyzję od briefu do pliku motywu oraz sprawdzić, czy osoba odpowiedzialna za treść potrafi później samodzielnie ją zmienić. Nazwy Atelier Północ, Studio Fala i Pracownia Punkt opisują fikcyjne realizacje, bez referencji prawdziwych klientów.

W folderze EXAMPLES otrzymujesz cały motyw blokowy wp-demo-lab, a nie zestaw luźnych wycinków. README prowadzi przez uruchomienie lokalnego WordPress, utworzenie stron, wybór Home oraz ustawienie nawigacji. Treści Gutenberg można wkleić do edytora kodu. Fragmenty pokazane w książce służą wyjaśnieniu mechanizmu; nie zastępują pełnych plików. Żadna obowiązkowa wtyczka ani nieistniejący blok projektu nie są potrzebne w tym etapie.

Ćwiczenie wykorzystuje theme.json w wersji 2 i standardowe bloki. Minimalny profil to WordPress 6.6 oraz PHP 8.2, jednak do własnego środowiska wybierz aktualne wspierane wydania. Zapisz ich konkretne wersje. W środowisku przygotowania publikacji nie było PHP ani Docker, dlatego wykonano wyłącznie jawnie opisane kontrole statyczne Python. Aktywacja motywu, poprawność edytora i testy przeglądarki pozostają zadaniami do wykonania na stagingu.

Pracuj w kolejności rozdziałów. Najpierw zatwierdź odbiorcę i zakres, potem uruchom bazę demonstracyjną, zaimportuj treść i przejdź przez protokół odbioru. Każdy prompt ma określony wynik oraz kontrolę. Zapisuj poprawki jako małe decyzje z uzasadnieniem. Profesjonalizm oznacza tutaj czytelny przekaz, edytowalność, dostępność i możliwość bezpiecznego odtworzenia strony po zmianie, a nie liczbę efektów czy automatyzacji.

Cel, który można sprawdzić

Brief zamienia ogólne oczekiwanie „chcę nową stronę” w zestaw decyzji, które da się ocenić podczas odbioru.

Zacznij od jednej sytuacji odbiorcy. Właściciel małej firmy szuka zespołu, który uporządkuje ofertę i odda stronę możliwą do aktualizowania. Potrzebuje wiedzieć, co obejmuje usługa, jak wygląda współpraca i jak przygotować opis potrzeby. Nie musi poznać szczegółów technologii przed zrozumieniem wartości oferty. To wyznacza kolejność informacji oraz sens głównego wezwania do działania.

Droga od potrzeby do kryterium

01 Odbiorca

Właściciel małej firmy szuka pomocy w uporządkowaniu oferty.

02 Zadanie

Rozumie usługę i wie, jaki opis potrzeby przygotować.

03 Zakres

Pięć stron; kontakt demonstracyjny; standardowe bloki.

04 Odbiór

Użytkownik odnajduje informację, właściciel edytuje treść.

W briefie zapisz rezultat: użytkownik odnajduje zakres usług i przechodzi do opisu kontaktu, a właściciel edytuje akapit bez pomocy programisty. Dopisz ograniczenia: pięć stron, treści fikcyjne, kontakt bez wysyłki, brak biblioteki plików i automatycznych publikacji. Ustal osobę zatwierdzającą treść oraz sposób zgłaszania błędów. Każde „później” powinno wskazywać etap rozwoju, aby nie stało się ukrytym obowiązkiem bieżącego wdrożenia.

Sprawdzenie briefu wymaga prostego scenariusza: poproś czytelnika o znalezienie usługi i wyjaśnienie następnego kroku własnymi słowami. Jeśli odpowiedź jest niejasna, popraw nazwy i teksty przed dekorowaniem układu. Właściciel powinien z kolei wskazać, gdzie zmieni tytuł, opis i przycisk. Wyniki takiej próby zapisuje się jako obserwacje, bez udawania badania konwersji. Otrzymujesz punkt odniesienia dla dalszych decyzji projektowych.

01 / ARCHITEKTURA INFORMACJI

Pięć adresów, pięć odpowiedzi

Mapa witryny jest mała, lecz każda strona ma własną rolę. Powtarzanie całej oferty na każdym adresie utrudnia orientację.

Home daje krótką odpowiedź, czym zajmuje się firma, oraz prowadzi do Usług i Realizacji. O firmie opisuje sposób pracy oraz zasady współpracy. Usługi pokazują zakres i rezultat poszczególnych propozycji. Realizacje uczą czytania portfolio przez problem, zakres i przekazane rozwiązanie. Kontakt wyjaśnia demonstracyjny charakter strony i przygotowanie briefu; nie sugeruje, że wiadomość została wysłana.

PROMPT DO CODEXA / Mapa informacji dla WP DEMO LAB

Wklej do Codexa razem ze wskazanymi danymi.

Cel: przygotuj architekturę pięciu stron WP DEMO LAB. Dane: fikcyjna firma projektująca strony i przepływy treści; strony Home, O firmie, Usługi, Realizacje, Kontakt; trzy realizacje Atelier Północ, Studio Fala, Pracownia Punkt; kontakt bez wysyłki. Wynik: dla każdej strony pytanie odbiorcy, 3–5 sekcji, jedno główne CTA i docelowy URL. Nie dodawaj usług, referencji ani danych kontaktowych. Oznacz niepewne informacje do decyzji człowieka.

KONTROLA: Porównaj pięć adresów z briefem. Każda sekcja odpowiada na inne pytanie, a CTA prowadzi do istniejącej strony. Usuń obietnice pozbawione danych.

Każda sekcja potrzebuje pytania, na które odpowiada. Zamiast nagłówka „Rozwiązania przyszłości” wybierz „Co obejmuje strona firmowa”. Zamiast pięciu identycznych przycisków „Więcej” użyj nazw działania zgodnych z celem linku. Krótsze etykiety są korzystne, jeżeli nadal można je zrozumieć poza otaczającym akapitem. Nawigacja pozostaje stała, a treść konkretnej strony rozwija jedynie jej własny temat.

Prompt pozwala uporządkować materiał przed budowaniem bloków. Dołącz rzeczywisty brief demonstracji, ponieważ sam model nie zna uzgodnionych ograniczeń. Nie przyjmuj nowych stron, formularzy ani obietnic jako automatycznej części projektu. Przeglądaj również kolejność sekcji: użytkownik powinien najpierw rozpoznać ofertę, potem poznać dowody i następny krok. Wątpliwość AI może być pytaniem do redaktora, lecz nie upoważnia do wymyślenia brakującego faktu.

01 / RAMY PROJEKTU

Kryteria i odpowiedzialność

Rozmowa o jakości staje się użyteczna, gdy wiadomo, kto podejmuje decyzję i co dokładnie ma zostać pokazane podczas odbioru.

Właściciel treści potwierdza nazwy usług i komunikaty. Projektant odpowiada za hierarchię informacji i zachowanie układu. Osoba wdrażająca sprawdza motyw, bazę, media i środowisko. Jedna osoba może pełnić kilka ról, ale odpowiedzialności pozostają osobne. AI może przygotować propozycję oraz wskazać ryzyko, natomiast nie staje się właścicielem decyzji ani źródłem faktów o firmie.

Zapisz kryteria w języku działania: wszystkie pięć pozycji menu prowadzi do właściwych stron; Home ma jeden czytelny tytuł główny; Kontakt nie zbiera danych; każda karta realizacji wskazuje fikcyjny charakter przykładu. Dodaj edycję przez właściciela, obsługę klawiaturą, kontrolę szerokości telefonu i odtworzenie kopii. Takie wymagania można pokazać. Określenia „nowoczesna” i „profesjonalna” wymagają przełożenia na widoczne zachowania i jakość treści.

Ustal również zawartość przekazania. Paczka obejmuje pliki motywu, treści stron, instrukcję konfiguracji, protokół stagingu oraz zapis ograniczeń. Sam ZIP motywu nie przenosi zapisanych stron, menu, mediów ani zmian Edytora witryny. Oddanie projektu powinno uwzględniać te warstwy, nawet jeśli ćwiczenie rozpoczyna się od pustej bazy. Dzięki temu kolejna osoba nie będzie odtwarzać konfiguracji z domysłów lub zrzutów ekranu.

Nie używaj deklaracji wzrostu sprzedaży jako kryterium bez pomiaru, okresu i punktu odniesienia. W tej demonstracji portfolio opisuje zakres prac, a nie wynik biznesowy. Po zakończeniu pierwszego etapu można zaplanować pomiar rzeczywistej witryny. Najpierw jednak trzeba stworzyć stronę o jasnym celu i stabilnej konfiguracji. Zmianę zakresu zapisuj wraz z wpływem na treść, testy i termin, zanim zaczniesz wpływać na wykonanie.

Co jest plikiem, a co rekordem

WordPress łączy pliki aplikacji z bazą danych. Motyw opisuje wygląd, ale nie jest kopią całej strony.

W plikach motywu przechowujemy style.css, theme.json, funkcje, szablony, części szablonów i wzorce. Baza zawiera strony z treścią blokową, ustawienia witryny, rekord nawigacji oraz zmiany zapisane w Edytorze witryny. Biblioteka mediów łączy rekordy załączników z plikami w uploads. Te warstwy współpracują podczas renderowania; żadna z nich samodzielnie nie odtwarza kompletnej witryny.

Warstwy odtwarzalnej witryny

01 Motyw

Szablony, style i wartości domyślne.

02 Baza

Strony, menu, ustawienia i nadpisanie edytora.

03 Media

Pliki uploads wraz z rekordami załączników.

04 Renderowanie

WordPress łączy właściwy szablon z treścią.

05 Kopia

Zgodny komplet warstw oraz zapis wersji środowiska.

To rozróżnienie ma praktyczny skutek. Kopiując folder wp-demo-lab do innej instalacji, otrzymujesz domyślny wygląd, lecz musisz jeszcze utworzyć strony i wybrać Home. Eksportując samą bazę, nie zabierasz zdjęć ani plików motywu. Git wersjonuje świadomie wybrane pliki i nie zastępuje kopii środowiska. Na staging przenosimy zgodny zestaw, a domeny zmieniamy narzędziem rozumiejącym format zapisanych danych.

W tomie 01 funkcje domenowe nie trafiają do functions.php. Realizacje są zwykłymi blokami w treści. Kolejny tom doda osobną wtyczkę, typ wdl_project i blok projektów. Dzięki temu zmiana motywu nie powinna usuwać modelu danych realizacji. Podczas rozwoju pytaj, czy dana rzecz określa wygląd, czy trwałe zachowanie biznesowe. Odpowiedź pomaga wybrać właściwe miejsce i ogranicza koszt późniejszej wymiany warstwy wizualnej.

02 / PLIKI PODSTAWOWE

Motyw ma rozpoznawalną strukturę

style.css zawiera nagłówek identyfikujący motyw. templates/index.html zapewnia domyślny szablon w strukturze motywu blokowego.

Folder wp-demo-lab umieść bezpośrednio w wp-content/themes. Nazwa motywu z nagłówka jest czytelną etykietą w panelu, natomiast nazwa katalogu identyfikuje lokalizację plików. Numer wersji pomaga śledzić paczkę i odświeżać style. Wymagania w nagłówku opisują dolną granicę przykładu; nie gwarantują, że każda przyszła wersja WordPress zachowa identyczny interfejs edycji. Wersje wykonawczego środowiska zapisuje się osobno.

KOD W PRAKTYCE / CSS / Nagłówek motywu

```
/*
Theme Name: WP DEMO LAB
Description: Edukacyjny motyw blokowy dla fikcyjnej firmy.
Version: 1.0.0
Requires at least: 6.6
Requires PHP: 8.2
Author: LAB PERCEPCJI
License: GNU General Public License v2 or later
License URI: https://www.gnu.org/licenses/gpl-2.0.html
Text Domain: wp-demo-lab
*/
:root { color-scheme: light; }
```

PLIK: wp-demo-lab/style.css

Fragment pełnego pliku w EXAMPLES. Sprawdzona zgodność tekstu; uruchomienie WordPress/PHP niewykonane.

W pełnym pakiecie są także front-page, page, single, archive i 404. Nagłówek oraz stopka pozostają w parts, a wzorec hero w patterns. Rozdzielenie umożliwia zmianę jednego elementu bez kopiowania całej witryny. Motyw nie zawiera rdzenia WordPress, bazy ani wtyczek kolejnych tomów. Nie próbuj instalować folderu EXAMPLES jako motywu; pliki style.css i theme.json muszą znajdować się bezpośrednio w jego właściwym katalogu.

Pokazany nagłówek jest fragmentem działającego zestawu plików, lecz aktywacja nie została wykonana w środowisku publikacji. Przed użyciem sprawdź paczkę lokalnym walidatorem, a potem otwórz panel własnego WordPress. Błędy PHP i edytora należy ocenić w rzeczywistym kontekście, nie na podstawie wyglądu tekstu w książce.

02 / HIERARCHIA WIDOKÓW

Szablon oddziela ramę od treści

page.html umieszcza tytuł i zawartość strony między wspólnymi częściami witryny. Treść usług nie powinna być wpisana na stałe w tym szablonie.

Blok post-title pobiera tytuł bieżącego rekordu i renderuje go jako H1. Blok post-content umieszcza zapisane bloki strony wewnątrz ramy. Dzięki temu O firmie, Usługi, Realizacje i Kontakt używają wspólnego układu, ale mają osobne teksty. Fragment rozpoczyna się już wewnątrz main; pełny plik zawiera również komentarz otwierający grupę i część nagłówkową. Nie wklejaj wycinka jako całego szablonu.

KOD W PRAKTYCE / HTML / Wnętrze page.html

```
<main id="wdl-main" class="wp-block-group wdl-main">
<!-- wp:post-title {"level":1} /-->
<!-- wp:post-content {"layout":{"type":"constrained"}} /-->
</main>
<!-- /wp:group -->
<!-- wp:template-part {"slug":"footer","area":"footer"} /-->
```

PLIK: wp-demo-lab/templates/page.html

Fragment pełnego pliku w EXAMPLES. Sprawdzona zgodność tekstu; uruchomienie WordPress/PHP niewykonane.

Home korzysta z front-page.html, który dodaje wzorzec hero z własnym H1. Treść pages/home.html zaczyna się więc od H2. Index jest szablonem awaryjnym, single obsługuje pojedynczy wpis, archive listę w kontekście archiwum, a 404 nieznany adres. Sama obecność plików nie dowodzi poprawnego wyboru szablonu. Na stagingu trzeba obejrzeć każdy kontekst i sprawdzić strukturę tytułów oraz nawigację.

Edycja szablonu w panelu nie zmienia automatycznie pliku w repozytorium. Zapisane nadpisanie może mieć pierwszeństwo podczas renderowania. Jeżeli zmiana page.html nie pojawia się na stronie, najpierw sprawdź warstwę Edytora witryny i pamięć podręczną, zamiast dokładać kolejne reguły CSS. To rozpoznanie oszczędza czas i zapobiega ukrywaniu rzeczywistej przyczyny.

Uruchom lokalne laboratorium

Plik `.wp-env.json` opisuje środowisko demonstracyjne. Jego wykonanie wymaga Docker; sama poprawność JSON nie uruchamia serwera WordPress.

Na własnym komputerze zainstaluj Node w aktualnym wspieranym wydaniu oraz Docker z obsługą Compose. Otwórz terminal w EXAMPLES i wykonaj `npm install`, następnie `npm run start`. Narzędzie `wp-env` powinno przygotować instalację dostępną pod `localhost:8888`. Lokalny użytkownik domyślny jest opisany w README; nie przenoś jego hasła do środowiska dostępnego z internetu. Dostarczona konfiguracja nie wymaga wtyczek.

KOD W PRAKTYCE / JSON / Konfiguracja wp-env

```
{
  "themes": [
    "./wp-demo-lab"
  ],
  "plugins": [],
  "phpVersion": "8.2",
  "port": 8888,
  "testsPort": 8889,
  "config": {
    "WP_DEBUG": true,
    "WP_DEBUG_LOG": true
  }
}
```

PLIK: `.wp-env.json`

Fragment pełnego pliku w EXAMPLES. Sprawdzona zgodność tekstu; uruchomienie WordPress/PHP niewykonane.

Pakiet używa aktualnego wydania narzędzia w chwili instalacji. Zachowaj powstały `package-lock.json` i zapis wersji; kolejne odtworzenia wykonuj przez `npm ci`. Wybór konkretnej wersji rdzenia ustaw według dokumentacji `wp-env`, jeśli potrzebujesz powtarzalnego testu. Nie traktuj zakresu „minimum 6.6” jako konfiguracji każdej instalacji. Zapisz również PHP, system oraz przeglądarkę, aby wynik próby można było zinterpretować.

W produkcji tej publikacji sprawdzono składnię konfiguracji w Python, lecz nie wykonano `npm install`, `wp-env start` ani aktywacji motywu. Gdy uruchomisz laboratorium, zachowaj komunikaty i datę. Uruchomienie zakończone błędem nie jest testem strony. Najpierw rozwiąż problem środowiska, potem sprawdzaj edytor i widoki. Staging powinien mieć odrębne konta oraz kontrolę dostępu.

03 / KONFIGURACJA STRON

Utwórz treść, zanim ocenisz wygląd

Aktywny motyw nie tworzy automatycznie pięciu rekordów stron. Ten krok wykonujesz świadomie w panelu demonstracyjnej instalacji.

Po aktywacji WP DEMO LAB ustaw polski język, tytuł witryny i właściwą strefę czasową. Utwórz Home ze slugiem home, O firmie ze slugiem o-firmie, Usługi jako usługi, Realizacje jako realizacje oraz Kontakt jako kontakt. W każdej stronie otwórz edytor kodu i wklej cały odpowiadający jej plik z pages, łącznie z komentarzami wp:. Wróć do widoku wizualnego, sprawdź bloki i dopiero wtedy zapisz lub opublikuj treść demo.

W Ustawienia → Czytanie wybierz statyczną stronę Home. Adres tej strony stanie się korzeniem witryny, mimo że rekord ma własny slug. Nie konfiguruj osobnej strony wpisów, ponieważ blog nie należy do przyjętego zakresu. Ustaw czytelne bezpośrednie odnośniki i sprawdź rzeczywiste URL po zapisaniu. Menu nie powinno odsyłać do pustych adresów tylko dlatego, że ich nazwy dobrze wyglądają w nagłówku.

Przy imporcie zwracaj uwagę na ostrzeżenie o nieprawidłowym bloku. Może oznaczać niekompletne kopiowanie, zmianę serializacji lub problem wersji. Nie naprawiaj go przez przypadkowe usuwanie treści. Zachowaj kopię, porównaj pełny plik i sprawdź wynik w aktualnej wersji edytora. Walidator Python kontroluje zbilansowanie komentarzy i dane atrybutów, ale nie odtwarza mechanizmu walidacji Gutenberg.

Zanim oceniasz odstępy, otwórz wszystkie strony z menu, także Kontakt i Realizacje. Potwierdź, że ich treść odpowiada wybranemu tytułowi i używa właściwego szablonu. Rozpoznanie pomyłonego rekordu lub Home ustawionego w złym miejscu jest prostsze na początku. W dzienniku zapisz sposób importu oraz wersję plików, aby później odróżnić zmiany treści od zmian motywu.

03 / PRACA Z AI

Codex porządkuje diagnozę środowiska

Dobry opis problemu zawiera polecenie, wersje i pełny komunikat. „WordPress nie działa” nie pozwala odróżnić błędu Docker od błędu motywu.

Gdy lokalne uruchomienie zawodzi, ogranicz materiał do danych potrzebnych do diagnozy. Podaj system, wersje Node i wp-env, stan Docker, katalog wykonania polecenia oraz dokładny komunikat. Nie dołączaj haseł, tokenów, plików z danymi klienta ani całej bazy. Jeżeli log zawiera sekret, usuń jego wartość przed przekazaniem i rozważ wymianę ujawnionego poświadczenia. W tej demonstracji nie ma potrzebnego klucza API.

PROMPT DO CODEXA / Diagnoza lokalnego startu

Wklej do Codexa razem ze wskazanymi danymi.

Cel: zdiagnozuj problem startu wp-env bez utraty danych. Dane: dołączam system, wersje Node/wp-env, stan Docker, katalog EXAMPLES, .wp-env.json i pełny zanonimizowany log polecenia npm run start. Wynik: maksymalnie trzy hipotezy w kolejności prawdopodobieństwa oraz bezpieczne kroki odczytu stanu z oczekiwanym rezultatem. Oddziel to, co wynika z logu, od założeń. Nie resetuj, nie usuwaj wolumenów i nie deklaruj uruchomienia WordPress.

KONTROLA: Sprawdź polecenia przed wykonaniem. Zapisz faktyczne wyjście każdego kroku; akceptuj hipotezę dopiero po zgodnym wyniku.

Poproś o rozdzielenie hipotez od potwierdzeń. Brak działającego silnika Docker, zajęty port i błąd PHP wymagają różnych kroków. Przydatny wynik to krótka kolejność diagnostyczna zaczynająca się od odczytu stanu, bez resetowania danych. Polecenie kasujące środowisko nie powinno być pierwszą odpowiedzią na niejasny błąd. Kopia bazy i uploads pozostaje warunkiem wszelkiego świadomego resetu.

Po każdej próbie zapisz rezultat. Jeśli serwer zaczyna odpowiadać, dopiero wtedy przejdź do aktywacji motywu i edycji stron. Prompt nie jest dowodem uruchomienia; dowodem jest wykonane polecenie, odpowiedź środowiska i widok sprawdzony przez człowieka. Codex może pomóc odczytać log, lecz nie powinien dopisywać pomyślnego wyniku działania, którego nie widział. Zachowaj tę zasadę również podczas późniejszego wdrażania.

Paleta i typografia mają jedno źródło

System wizualny działa najlepiej, gdy nazwy kolorów, szerokości i rozmiary typografii są wspólne dla frontendu oraz edytora.

theme.json w tym motywie ma wersję 2. To świadoma decyzja wspólnego kontraktu, a nie deklaracja najnowszego formatu. Ustawienia określają dostępne presety, natomiast styles opisuje zastosowanie ich do witryny i elementów. Las, papier i atrament tworzą podstawowe pary tekstu oraz tła. Mosiądz pozostaje akcentem dekoracyjnym, ponieważ jego kontrast z papierem jest zbyt niski dla zwykłego tekstu.

KOD W PRAKTYCE / JSON / Początek wspólnej konfiguracji

```
{
  "$schema": "https://schemas.wp.org/wp/6.5/theme.json",
  "version": 2,
  "settings": {
    "appearanceTools": true,
    "useRootPaddingAwareAlignments": true,
    "layout": {
      "contentSize": "760px",
      "wideSize": "1160px"
    },
    "color": {
      "custom": false,
      "defaultPalette": false,
      "palette": [
        {
          "slug": "forest",
```

PLIK: wp-demo-lab/theme.json

Fragment pełnego pliku w EXAMPLES. Sprawdzona zgodność tekstu; uruchomienie WordPress/PHP niewykonane.

Szerokość treści 760 pikseli ogranicza długość akapitów, a wideSize 1160 pikseli daje miejsce na szersze kompozycje. To wartości projektowe, które trzeba sprawdzić z rzeczywistym tekstem, powiększeniem i urządzeniem. Typografia wykorzystuje fonty systemowe, więc motyw nie wymaga pobierania plików z zewnętrznego serwera. Rozmiar hero reaguje na szerokość przez clamp, zachowując dolną i górną granicę.

Pokazany początek pliku nie jest samodzielnym JSON do wklejenia. Pełna konfiguracja zawiera paletę, odstępy, style elementów i definicje części. Zmieniaj jeden parametr, sprawdzaj wynik w edytorze oraz na stronie, a potem zapisuj decyzję. Globalne style zapisane w bazie mogą nadpisać domyślny motyw, dlatego niespójności wymagają porównania obu źródeł.

04 / UI I HIERARCHIA

Interfejs prowadzi do decyzji

Najważniejszy komunikat musi być rozpoznawalny bez przewijania przez serię efektownych, lecz niejasnych sekcji.

Home zaczyna się od propozycji „Strona, którą potrafisz prowadzić” i wyjaśnienia odbiorcy. Przycisk prowadzi do Usług. Kolejne sekcje pokazują zakres pomocy, fikcyjne realizacje i opis następnego kroku. Układ ma wspierać tę kolejność. Gdy wszystkie elementy są jednakowo duże i intensywne, użytkownik traci wskazówkę, od czego zacząć. Zwiększenie odstępu bywa skuteczniejsze niż dodanie ramki, cienia lub kolejnej ikony.

PROMPT DO CODEXA / Przegląd hierarchii Home

Wklej do Codexa razem ze wskazanymi danymi.

Cel: oceń hierarchię Home WP DEMO LAB i zaproponuj trzy konkretne poprawki. Dane: dołączam pages/home.html, tekst hero, theme.json oraz opis lub zrzut bieżącego widoku. Wynik: dla każdej uwagi wskaż element, problem odbiorcy, zmianę i sposób kontroli. Zachowaj pięć stron, fikcyjne realizacje oraz CTA do istniejących URL. Nie dopisuj wyników badań, konwersji ani nowych integracji. Oddziel obserwację od hipotezy.

KONTROLA: Zastosuj jedną zmianę naraz. Sprawdź, czy odbiorca szybciej wyjaśnia ofertę i odnajduje przejście do Usług; zapisz rzeczywistą obserwację.

Porównuj warianty z tym samym tekstem. Inaczej trudno ustalić, czy poprawę przyniósł układ, czy krótszy nagłówek. Oceniaj czytelność, rozpoznawalność głównego działania i spójność sekcji. AI może zaproponować kolejność lub wskazać miejsca przeciążenia, ale nie zna faktycznej reakcji odbiorców. Nie prezentuj jego opinii jako wyniku badania użytkowników. Krótka próba z człowiekiem daje inne informacje niż analiza tekstu.

Przed zmianą systemu kolorów sprawdź rzeczywistą parę, rozmiar tekstu oraz stan hover i focus. Wyrazistość wizualna nie zastępuje kontrastu. Nie tworzymy tu ciemnej sekcji, której elementy wymagają nieprzetestowanych wyjątków; wszystkie podstawowe widoki opierają się na jasnym papierze. Jeżeli dodasz nowy wariant, obejmij go osobnym testem czytelności, klawiatury oraz zachowania na małej szerokości.

04 / DOŁĄCZANIE STYLÓW

CSS i edytor korzystają z tej samej paczki

functions.php ma niewielkie zadanie: dołączyć arkusz motywu. Nie przechowuje formularzy, modeli danych ani sekretów integracji.

Hook `enqueue_block_assets` służy tutaj do wspólnego dołączenia arkusza dla bloków na frontendzie i w kontekście edytora. Używamy `get_stylesheet_uri` zamiast ręcznego składania adresu pliku. Numer wersji pochodzi z nagłówka motywu, co pozwala zmienić identyfikator zasobu po aktualizacji paczki. Zadbaj o podniesienie wersji przy wydaniu, zamiast wymuszać kolejne odświeżenia tylko w swojej przeglądarce.

KOD W PRAKTYCE / PHP / Dołączenie arkusza

```
<?php
/** Styles shared by the frontend and block editor. */
if ( ! defined( 'ABSPATH' ) ) {
    exit;
}
function wdl_enqueue_styles() {
    $theme = wp_get_theme();
    wp_enqueue_style(
        'wp-demo-lab',
        get_stylesheet_uri(),
        array(),
        $theme->get( 'Version' )
    );
}
add_action( 'enqueue_block_assets', 'wdl_enqueue_styles' );
```

PLIK: wp-demo-lab/functions.php

Fragment pełnego pliku w EXAMPLES. Sprawdzona zgodność tekstu; uruchomienie WordPress/PHP niewykonane.

Podstawowe style typograficzne i kolory opisuje theme.json. W CSS zostają zachowania specyficzne dla tego przykładu: siatka kart, przejście na jedną kolumnę, skip link i obrys fokusu. Taki podział ułatwia diagnozę. Jeżeli edytor wygląda inaczej niż frontend, porównaj załadowane zasoby, kontekst iframe i zapisane style globalne. Nie zakładaj, że kolejny selektor z `important` rozwiązuje źródło problemu.

Fragment odpowiada pełnemu functions.php. Python potwierdził obecność pliku, ale nie wykonano sprawdzenia składni PHP ani renderowania przez WordPress. Na własnym środowisku uruchom `php -l`, aktywuj motyw i sprawdź log. Następnie porównaj edytor i widok publiczny na tych samych blokach. Błąd runtime wymaga rzeczywistego komunikatu, nie oceny po wyglądzie kodu.

Blok, wzorzec i część szablonu

Trzy mechanizmy mają różne zadania. Ich rozróżnienie pomaga utrzymać wspólny układ i nie zmieniać przypadkiem całej witryny.

Blok jest jednostką treści lub funkcji: akapit, nagłówek, grupa, przycisk albo nawigacja. Wzorzec dostarcza ułożoną kompozycję bloków jako punkt startowy. Część szablonu przechowuje fragment wspólnej ramy, na przykład nagłówek lub stopkę. W WP DEMO LAB zwykłe treści stron używają podstawowych bloków, hero jest wzorcem, a menu i tytuł witryny mieszkają w części header.

Dobór elementu według zasięgu

01 Pojedyncza treść

Akapit i nagłówek: standardowe bloki.

02 Powtarzalny układ

Hero: wzorzec z ułożonych bloków.

03 Wspólna rama

Header i footer: części szablonu.

04 Dane domenowe

Portfolio z rekordami: wtyczka w kolejnym tomie.

Zwykłe wstawienie niesynchronizowanego wzorca do treści daje kopię, którą można lokalnie edytować. Zmiana pliku wzorca nie aktualizuje wszystkich takich kopii. Wzorce synchronizowane tworzone w edytorze wymagają osobnego zarządzania rekordem; nie są tym samym co plik hero.php. Część szablonu jest współdzielona przez odwołanie, dlatego jej modyfikacja wpływa na wiele widoków. Zawsze określ oczekiwany zasięg zmiany przed zapisaniem.

W tym tomie nie rejestrujemy własnego bloku realizacji. Trzy karty są zwykłymi grupami i nagłówkami w pages/realizacje.html. Taki początek jest kompletny bez przyszłej wtyczki. Rozszerzenie portfolio przyjdzie później jako funkcja domenowa, z osobnymi rekordami oraz dynamicznym renderowaniem. Zachowanie nazw projektów zapewnia ciągłość serii, lecz nie wymaga przedwczesnego odwołania do niedostępnej funkcji.

05 / WZORCE

Hero jest wzorcem z nagłówkiem pliku

Katalog patterns pozwala dostarczyć nazwany układ. Plik PHP zawiera metadane i markup blokowy, a nie cały silnik strony.

Nagłówek wzorca określa czytelną nazwę, slug wp-demo-lab/hero oraz kategorię featured. Front-page odwołuje się do tego slugu. Pełny plik dodaje H1, akapit oraz przycisk prowadzący do Usług. Adres tego przycisku wykorzystuje home_url i esc_url, dzięki czemu nie wpisujemy domeny instalacji na stałe. Pozostałe demonstracyjne linki od korzenia wymagają sprawdzenia przy instalacji w podkatalogu.

KOD W PRAKTYCE / PHP / Metadane hero.php

```
<?php
/**
 * Title: WP DEMO LAB – wprowadzenie
 * Slug: wp-demo-lab/hero
 * Categories: featured
 * Description: Nagłówek Home i przejście do usług.
 */
if ( ! defined( 'ABSPATH' ) ) {
    exit;
}
?>
```

PLIK: wp-demo-lab/patterns/hero.php

Fragment pełnego pliku w EXAMPLES. Sprawdzona zgodność tekstu; uruchomienie WordPress/PHP niewykonane.

Hero nie powinno obiecywać rzeczy, których oferta nie potwierdza. Tekst mówi o przejrzystej stronie i przepływie pracy, a nie o gwarantowanym zysku. Gdy zmienisz nagłówek, odczytaj go razem z akapitem i CTA. Użytkownik musi rozpoznać, kto otrzyma pomoc i do czego prowadzi działanie. Sam efektowny slogan nie wyjaśnia usługi. Krótszy tekst ułatwia też zawijanie na telefonie oraz pracę z powiększeniem.

Fragment pokazuje metadane i ochronę przed bezpośrednim wykonaniem pliku. Właściwa kompozycja znajduje się dalej w EXAMPLES. Jeżeli w edytorze brak wzorca, sprawdź nagłówek, położenie pliku i aktywny motyw. Nie rejestruj tej samej kompozycji drugi raz w functions.php; dwie konkurujące definicje utrudniają ustalenie, z którego pliku pochodzi widoczna treść.

05 / CODEX I BLOKI

Zmiana komponentu ma mały zakres

Codex pomaga zmieniać układ, kiedy otrzymuje pełny plik, oczekiwane zachowanie i ograniczenia. Przypadkowa przebudowa wszystkich stron utrudnia ocenę.

Wybierz konkretny komponent, na przykład karty usług. Podaj obecny markup i style oraz problem: za długie opisy, nieczytelna kolejność lub brak miejsca przy małej szerokości. Wynik powinien obejmować minimalny diff i uzasadnienie każdej zmiany. Nie proś jednocześnie o nowe portfolio, formularz i bibliotekę. Te funkcje mają osobne odpowiedzialności i wymagania, których pierwszy motyw nie realizuje.

PROMPT DO CODEXA / Minimalna poprawka kart

Wklej do Codexa razem ze wskazanymi danymi.

Cel: popraw układ kart usług bez zmiany treści i bez własnych bloków. Dane: pages/uslugi.html, style.css, theme.json; problem opisuję osobno. Wynik: minimalny diff tylko dla potrzebnych plików oraz krótka lista zachowań do sprawdzenia. Zachowaj komentarze Gutenberg, strukturę H2/H3, klasy wdl-grid i wdl-card oraz standardowe bloki rdzenia. Nie dodawaj bibliotek, pluginów ani bloku wp-demo-lab/projects.

KONTROLA: Sprawdź zbilansowanie bloków, import do edytora bez ostrzeżeń, jedną kolumnę na telefonie i zgodność treści. Test WordPress wykonaj dopiero na własnej instalacji.

Markup Gutenberg składa się z komentarzy z atrybutami oraz odpowiadającego im HTML. Zmiana nagłówka czy klasy musi zachować zgodność obu części. Użycie losowego HTML wewnątrz bloku nie daje pewności, że edytor uzna zapis za prawidłowy. Po poprawce uruchom walidator statyczny, wklej pełną treść do własnego WordPress, przejdź do widoku wizualnego i zapisz. Sprawdź, czy układ pozostaje edytowalny.

Oceniaj również zasięg CSS. Klasa wdl-card ma dotyczyć kart demonstracyjnych, a nie każdego akapitu witryny. Globalna reguła dla wszystkich elementów może zmienić stopkę, nawigację lub panel mobilny. Nie traktuj poprawnego podglądu jednego komponentu jako dowodu dla całej strony. Dobrze zawężona zmiana ogranicza konieczne sprawdzenia, ale nadal wymaga kontroli na docelowym widoku i w edytorze.

Treść Home, O firmie i Usług

Pierwsze trzy widoki budują ciąg: rozpoznanie oferty, zrozumienie sposobu pracy i ocena konkretnego zakresu.

Home nie musi streszczać wszystkiego. Wzorzec hero podaje główną obietnicę, a treść strony rozwija potrzeby małej firmy, obszary pomocy, przykłady i następny krok. Widok służy orientacji. O firmie pokazuje sposób działania: ustalenie celu, zebranie materiałów, strukturę, staging i odbiór. W tej demonstracji nie dopisujemy biografii zespołu, dat założenia ani liczby klientów, ponieważ brak danych, które je potwierdzają.

Usługi są opisane przez rezultat. Strona firmowa daje pięć widoków i możliwość edycji. Porządek w treści prowadzi do struktury oraz zasad redakcyjnych. Rozwój z AI oznacza przygotowanie kontrolowanych szkiców, a nie uruchomioną automatyczną publikację. Rozdziel zakres bieżącego etapu od możliwości kolejnych tomów. Dzięki temu odbiorca nie pomyli edukacyjnego przykładu z gotową ofertą działającego systemu integracji.

Wszystkie treści znajdują się w oddzielnych plikach pages. Właściciel edytuje zapis strony, natomiast projektant utrzymuje szablon. Jeśli potrzebujesz poprawić jeden opis usługi, nie zmieniaj page.html. Jeśli chcesz dodać wspólne odstępy wokół wszystkich stron, nie kopiuj tej samej reguły do pięciu rekordów. Ten podział pomaga zachować możliwość redakcji i ogranicza liczbę miejsc, które trzeba później zsynchronizować.

Po imporcie odczytaj stronę na głos. Sprawdź, czy każde zdanie przekazuje nową informację, a nagłówek odpowiada następującemu akapitowi. Usuń puste zapewnienia, takie jak „najwyższa jakość dla każdego”. Zastąp je działaniem lub zakresem, który można obejrzeć. Konsekwentny język powinien występować także w menu i na przyciskach. Słowo „Kontakt” zawsze prowadzi do jawnej demonstracji, nie do formularza udającego wysłanie wiadomości.

06 / PROMPT REDAKCYJNY

AI redaguje wyłącznie otrzymane fakty

Kontrola treści zaczyna się przed generowaniem. Model powinien wiedzieć, które informacje są zatwierdzone i gdzie nie wolno mu uzupełniać luk.

Przeznacz AI listę faktów i oddziel ją od oczekiwanego tonu. W tym przykładzie zatwierdzone są nazwa fikcyjnej firmy, zakres pięciu stron, trzy fikcyjne realizacje oraz brak wysyłki na Kontakt. Ton może być spokojny i rzeczowy. Nie ma zatwierdzonych cen, adresu, godzin pracy ani statystyk efektów. Te informacje nie powinny pojawić się tylko dlatego, że podobne strony zwykle je zawierają.

PROMPT DO CODEXA / Szkic treści bez nowych obietnic

Wklej do Codexa razem ze wskazanymi danymi.

Cel: zredaguj stronę Usługi w rzeczowym języku. Dane: zatwierdzone opisy z [pages/uslugi.html](#) i brief pięciu stron WP DEMO LAB. Wynik: tytuły sekcji, krótkie akapity, etykieta CTA oraz osobna lista wykorzystanych faktów. Zachowaj fikcyjny charakter firmy; kontakt nie wysyła wiadomości. Nie dodawaj cen, terminów, adresów, certyfikatów, statystyk ani gwarancji. Braki oznacz „do decyzji redaktora”.

KONTROLA: Porównaj każde twierdzenie z danymi wejściowymi. Sprawdź, czy tekst określa rezultat i zakres, nie sugeruje działającej integracji i pasuje do bloków strony.

Poproś o szkic i listę użytych faktów. Pozwala to porównać tekst z materiałem wejściowym, ale nadal potrzebny jest przegląd człowieka. Nie wszystko, co model oznaczy jako fakt, rzeczywiście nim jest. Zwróć uwagę na ukryte obietnice: „odpowiadamy jutro”, „zwiększymy sprzedaż” lub „pełna zgodność z WCAG”. Każda z nich wymaga procesu i dowodów. W demonstracji najbezpieczniej opisać zakres oraz zadania do wykonania.

Redaguj najpierw tekst bez znaczników, dopiero potem przypisz go do bloków. Ułatwia to ocenę długości i hierarchii. Po wstawieniu do Gutenberg sprawdź zawijanie, nagłówki i etykiety linków. Skrócenie akapitu nie może usuwać informacji, że firma i realizacje są fikcyjne. Zachowaj to oznaczenie przy portfolio również wtedy, gdy karty będą później zasilane osobnymi rekordami wtyczki projektów.

06 / UCZCIWY ZAKRES

Realizacje i Kontakt budują zaufanie

Portfolio może być konkretne bez wymyślonych wyników. Kontakt może być użyteczny, nawet gdy pierwszy etap nie obejmuje zbierania danych.

Atelier Północ opisuje uporządkowanie oferty pracowni wewnątrz: mapę strony, proces i przejście do kontaktu. Studio Fala pokazuje stronę studia kreatywnego z trzema obszarami usług oraz selekcją prac. Pracownia Punkt prezentuje ofertę małej pracowni rzemieślniczej i instrukcję aktualizacji. Karty mówią o problemie i zakresie, nie o liczbie leadów czy przychodzie. Nad sekcją znajduje się wyjaśnienie, że wszystkie przykłady są fikcyjne.

To świadomie statyczna treść. Nie ma jeszcze typów wpisów, pól meta ani wywołań REST portfolio. W tomie 02 zachowamy nazwy i przejdziemy do rekordów wdl_project. Dzięki takiej kolejności odbiorca rozumie najpierw redakcyjne zadanie karty, a dopiero później jej model danych. Nie trzeba instalować przyszłej wtyczki, aby motyw pierwszego tomu dał kompletną, czytelną witrynę.

Kontakt ma cztery funkcje: ujawnia demonstrację, wyjaśnia przygotowanie opisu potrzeby, zapowiada osobny etap formularza i daje drogę powrotu do Usług. Nie ma telefonu, mailto, pól ani pozornego przycisku „Wyślij”. Użytkownik nie powinien zastanawiać się, czy wiadomość trafiła do firmy. Jeśli w prawdziwym wdrożeniu dodasz kanał kontaktu, musisz zapewnić jego działanie, właściciela odpowiedzi i właściwy opis przetwarzania danych.

Przegląd tych stron wykonaj także jako osoba, która nie czytała książki. Otwórz portfolio bezpośrednio z adresu, przeczytaj jedną kartę i przejdź do Kontakt. Czy oznaczenie fikcji pozostaje zrozumiałe? Czy następny krok nie obiecuje niemożliwej akcji? Taka próba pomaga odnaleźć niedopowiedzenia, których autor przyzwyczajony do projektu już nie zauważa. Zapisz wynik jako obserwację własnej próby, bez udawania testu szerszej grupy użytkowników.

Media wymagają planu przed importem

Pierwszy motyw działa bez zdjęć. Dodanie obrazu jest decyzją o treści, dostępności i transferze, a nie obowiązkowym ozdobnikiem.

Wybierz obraz, który pomaga zrozumieć usługę lub zakres realizacji. Ustal prawa do użycia i zachowaj informację o źródle. Nie przypisuj wygenerowanej wizualizacji do prawdziwej realizacji ani nie przedstawiaj fikcyjnego wnętrza jako dokumentacji klienta. Przy ilustracji edukacyjnej podpis powinien wyjaśniać jej charakter. Unikaj grafiki z małym tekstem, ponieważ trudniej ją odczytać, powiększyć i dostosować do urządzenia.

PROMPT DO CODEXA / Karta publikacji obrazu

Wklej do Codexa razem ze wskazanymi danymi.

Cel: przygotuj kartę obrazu do biblioteki mediów, bez zgadywania niewidocznych szczegółów. Dane: dołączony obraz, miejsce użycia, funkcja, informacja o prawach i planowany rozmiar. Wynik: nazwa pliku, tytuł, propozycja podpisu, opis źródła, alt lub uzasadnienie pustego alt oraz pytania wymagające decyzji. Oznacz fikcyjną/generowaną ilustrację, jeśli wynika to z danych. Nie wymyślaj nazw osób ani klientów.

KONTROLA: Obejrzyj obraz i potwierdź każde twierdzenie. Sprawdź prawa, funkcję alt w tym kontekście, kadrowanie oraz rozmiar pobieranego pliku.

Pola biblioteki mediów mają różne role. Tytuł ułatwia identyfikację załącznika, podpis pojawia się przy obrazie, opis może przechowywać kontekst, a alt przekazuje funkcję obrazu w miejscu użycia. Te pola nie zastępują się wzajemnie. Ten sam obraz może potrzebować innego alt w innym kontekście. Dekoracja powinna mieć pusty alt, a nie opis „ładne tło”. Obraz będący linkiem wymaga zrozumiałego celu działania.

Przygotuj odpowiednie wymiary przed wysłaniem i sprawdź responsywne warianty generowane przez WordPress. Nie dodawaj zdjęcia o wielkim transferze tylko po to, by wyświetlić je w małej karcie. Po imporcie obejrzyj kadrowanie na telefonie, podpis oraz rzeczywiste atrybuty na froncie. Propozycję opisu AI zatwierdza osoba, która widzi obraz i zna jego zadanie; sam tytuł pliku nie wystarcza do poprawnego alt.

07 / RESPONSYWNOŚĆ

Układ zmienia się razem z przestrzenią

Trzy kolumny kart są wygodne na szerokim ekranie. Węższy widok potrzebuje jednej kolumny i kolejności zgodnej z dokumentem.

Klasy `wdl-grid` i `wdl-card` działają na zwykłych grupach Gutenberg. Media query zmienia liczbę kolumn przy 48 rem, a wiersz nagłówka układu pionowo. To prosta strategia dla małego przykładu. Punkt przejścia trzeba ocenić z docelową długością tekstu, szerokością menu i powiększeniem. Rozmiary konkretnych urządzeń nie są jedyną przyczyną zmiany dostępnego miejsca.

KOD W PRAKTYCE / CSS / Siatka i przejście mobilne

```
.wdl-grid { display: grid; grid-template-columns: repeat(3, 1fr);
  gap: 1.5rem; }
.wdl-card { border: 1px solid #123c32; padding: 1.5rem; }
.wdl-header-row { gap: 1rem; }
.wp-block-navigation a { padding-block: .65rem; }
.wp-block-button__link { min-height: 44px; }
.wp-block-post-content { overflow-wrap: anywhere; }
@media (max-width: 48rem) {
  .wdl-grid { grid-template-columns: 1fr; }
  .wdl-header-row { flex-direction: column; align-items: start; }
}
```

PLIK: `wp-demo-lab/style.css`

Fragment pełnego pliku w EXAMPLES. Sprawdzona zgodność tekstu; uruchomienie WordPress/PHP niewykonane.

Zachowujemy kolejność bloków w kodzie. Nie przestawiamy kart wizualnie tak, aby klawiatura lub czytnik ekranu podążały inną ścieżką. Tekst ma naturalnie zawijać się, a obrazy zachowywać proporcje. Nie ustawiaj stałej wysokości karty, jeśli dłuższy opis może zostać obcięty. Przyciski powinny pozostawać widoczne także wtedy, gdy użytkownik zwiększa rozmiar tekstu lub korzysta z innej czcionki systemowej.

Fragment pochodzi z pełnego arkusza. Po zmianie sprawdź minimum 320 CSS pikseli, typowy telefon, tablet i szeroki ekran, a także zoom. Brak poziomego przewijania przy jednym tekście nie gwarantuje go po późniejszej redakcji. Zapisz konkretne szerokości i przypadki, zamiast ogólnego zapewnienia, że witryna „jest responsywna”.

07 / PRÓBA UKŁADU

Obserwuj stronę z rzeczywistą treścią

Test responsywności dotyczy informacji i działań. Zrzut pustych kart nie mówi, czy użytkownik odnajdzie ofertę po zwiększeniu tekstu.

Zacznij od widoku wąskiego i przeczytaj całą stronę. Czy nagłówek mieści się bez przycięcia? Czy przycisk zachowuje zrozumiałą etykietę? Czy menu można otworzyć oraz zamknąć? Następnie zwiększ rozmiar tekstu i sprawdź długi opis usługi, dłuższą nazwę realizacji oraz link. Takie dane są bardziej użyteczne niż idealnie dopasowane jednozdaniowe próbki, które nie ujawniają ograniczeń komponentu.

Sprawdź również rytm pionowy. Duży odstęp może odciąć nagłówek od jego treści, a zbyt mały zlać dwie sekcje. Na telefonie trzy karty stają się długą sekwencją; rozpoznawalne tytuły pomagają ją przeskanować. Nie ukrywaj podstawowych informacji wyłącznie po to, aby ekran wyglądał krócej. Jeżeli dodajesz rozwijane elementy, ich obsługa, stan i dostępność wymagają odrębnego sprawdzenia.

Wydajność mediów oceń na faktycznie pobieranych plikach. WordPress może przygotować srcset, ale jakość wyniku zależy także od źródła, ustawień i miejsca użycia. Bez pomiaru nie podajemy wyniku PageSpeed ani czasu ładowania. Można zapisać widoczny problem i plan pomiaru, lecz nie zamieniać przypuszczenia w benchmark. W tym pakiecie brak obrazów ogranicza zależności i pozwala skupić się na treści.

Po korekcie przejdź ponownie przez menu i CTA na dwóch istotnych szerokościach. Nowa reguła dla siatki może nie dotyczyć nagłówka, ale zmieniony padding globalny już tak. Sprawdzenie powinno odpowiadać zakresowi zmiany. Zachowaj notatkę: szerokość, powiększenie, przeglądarka, obserwacja i decyzja. Taka informacja jest lepszą podstawą odbioru niż nieprecyzyjne „wygląda dobrze na moim laptopie”.

Focus musi być widoczny i użyteczny

Obrys fokusu pokazuje miejsce działania klawiatury. Usunięcie go bez równoważnego rozwiązania utrudnia korzystanie z linków i menu.

Arkusze używa `focus-visible` dla podstawowych kontrolki. Ciemny obrys i jasny pierścień pomagają odnaleźć element na jasnym tle. W tym motywie to rozwiązanie bazowe; nowy ciemny wariant sekcji wymaga ponownej oceny. Sama reguła nie gwarantuje, że obrys nie zostanie obcięty przez overflow lub zasłonięty nakładką. Musisz zobaczyć rzeczywisty stan w przeglądarce.

KOD W PRAKTYCE / CSS / Bazowy obrys fokusu

```
a:focus-visible, button:focus-visible, input:focus-visible,
textarea:focus-visible, select:focus-visible {
  outline: 3px solid #123c32;
  outline-offset: 4px;
  box-shadow: 0 0 0 4px #faf8f1;
}
```

PLIK: wp-demo-lab/style.css

Fragment pełnego pliku w EXAMPLES. Sprawdzona zgodność tekstu; uruchomienie WordPress/PHP niewykonane.

Skip link znajduje się na początku nagłówka i po otrzymaniu fokusu prowadzi do `wdl-main`. Sprawdź Tab, Enter i dalsze przechodzenie po treści. Następnie przejdź wszystkie pozycje menu oraz CTA, wróć przez Shift+Tab i otwórz menu mobilne. Oceń zamknięcie Escape i powrót do kontrolki otwierającej. Kolejność powinna zgadzać się z układem i nie prowadzić do pułapki.

To zadania do wykonania na stagingu, nie rezultaty produkcji publikacji. Python sprawdził obecność linku i identyfikatora, ale nie symuluje przenoszenia fokusu ani zachowania mobilnej nawigacji WordPress. Zapisz faktyczny wynik każdego działania. Gdy problem występuje tylko na jednej szerokości, dołącz ją do opisu błędu wraz z przeglądarką i krokami odtworzenia.

Kontrast, nagłówki i kontekst

Dostępność jest zestawem sprawdzanych zachowań i właściwości treści. Wybór spokojnej palety nie wystarczy do stwierdzenia zgodności.

Obliczenia dla podstawowych par motywu dały 11,50:1 dla lasu na papierze oraz 12,55:1 dla atramentu na papierze. Mosiądz na papierze ma około 2,97:1. Dlatego nie stosujemy go jako zwykłego tekstu ani informacji wymagającej rozpoznania. WCAG określa dla zwykłego tekstu minimum 4,5:1, a dla odpowiednio dużego tekstu 3:1. Osobno ocenia się elementy interfejsu i focus. Zaokrąglony wynik nie powinien maskować wartości poniżej progu.

Przejrzyj drzewo nagłówków. Home ma H1 w hero, inne strony H1 z post-title. Sekcje otrzymują H2, a karty w ich obrębie H3. Nie wybieraj H3 tylko dlatego, że jego rozmiar dobrze pasuje do kompozycji; wygląd można zmienić osobno. Nagłówek opisuje strukturę dokumentu. Sprawdź również znaczniki header, main i footer oraz to, czy treść nie ma kilku konkurujących obszarów głównych.

Link powinien wyjaśniać działanie. „Poznaj zakres usług” ma większą wartość niż powtarzane „kliknij tutaj”. Informacja o fikcyjnych realizacjach pozostaje tekstem, a nie wyłącznie kolorem czy ikoną. Dodając obraz, opisz jego funkcję przez alt. Dodając interakcję, zapewnij nazwę, stan i obsługę klawiaturą. Standardowy blok pomaga, ale konfiguracja, treść i zewnętrzne style mogą zmienić wynik.

Kontrola kontrastu w Python dotyczy zadanych kolorów, nie całego ekranu. Nie obejmuje zdjęć, przezroczystości, gradientów ani wszystkich stanów. Lista podstawowych prób nie jest formalnym audytem WCAG. W protokole oddziel wykonane pomiary od zadań do sprawdzenia i jasno zapisuj brakujące dowody. Uczciwy zakres raportu ułatwia późniejsze poprawki oraz rozmowę z osobą odbierającą stronę.

08 / PRZEGLĄD Z CODEXEM

Zamień checklistę w scenariusze

AI może pomóc ułożyć próbę dostępności, ale wynik obserwacji pochodzi z działania w rzeczywistej stronie i przeglądarce.

Daj Codexowi konkretne widoki oraz zakres funkcji. W naszym przykładzie są to menu mobilne, skip link, nagłówki, linki i statyczne karty. Kontakt nie ma formularza, więc nie trzeba udawać testu jego walidacji. Model może wskazać, jakie scenariusze będą potrzebne po dodaniu formularza, lecz oddziel je od obecnego protokołu. Nadmiernie szeroka lista utrudnia wykonanie podstawowych prób.

PROMPT DO CODEXA / Scenariusze dostępności pięciu stron

Wklej do Codexa razem ze wskazanymi danymi.

Cel: przygotuj protokół ręcznych testów dostępności WP DEMO LAB. Dane: header.html, page.html, front-page.html, style.css i pięć plików treści; brak formularza oraz wysyłki. Wynik: 8–12 scenariuszy z warunkiem początkowym, krokami, oczekiwanym zachowaniem i polem wyniku. Obejmij skip link, Tab/Shift+Tab, menu mobilne i Escape, H1/H2/H3, zoom, reflow i rzeczywiste pary kolorów. Oddziel przegląd kodu od testu w przeglądarce.

KONTROLA: Wykonaj scenariusze na stagingu. Nie wpisuj OK bez obserwacji, nie deklaruj zgodności całej witryny na podstawie samego CSS.

Scenariusz zawiera warunek początkowy, działanie i oczekiwane zachowanie. „Sprawdź dostępność menu” jest zbyt ogólne. Lepiej określić wąski widok, wejście Tab na przycisk, otwarcie, przejście po linkach, Escape i powrót fokusu. Tester może wtedy zapisać miejsce różnicy. Powiększenie i kontrast mają analogicznie wskazywać konkretny element lub stan, zamiast ograniczać się do ogólnego wrażenia.

Nie przyjmuj automatycznej oceny „zgodne z WCAG”, nawet gdy model nie dostrzega problemów w kodzie. Część wymagań zależy od treści, interakcji i kontekstu. Statyczny przegląd może znaleźć ryzyko lub zasugerować pytanie. Właściwy raport opisuje wykonane działania, zaobserwowane zachowania i brakujące sprawdzenia. Tak przygotowany materiał nadaje się do pracy nad poprawkami, ponieważ nie ukrywa granic dowodów.

Staging jest miejscem uzgadniania zmian

Osobne środowisko pozwala obejrzeć treść, sprawdzić nadpisanie i wykonać próbę odtworzenia przed decyzją o produkcji.

Staging ma własną instalację i bazę, HTTPS oraz ograniczenie dostępu na poziomie serwera. Prośba do wyszukiwarek o nieindeksowanie jest dodatkowym ustawieniem, nie kontrolą dostępu. Nie przenoś lokalnego hasła wp-env do publicznego środowiska. Każda osoba powinna mieć własne konto z uprawnieniami potrzebnymi do zadania. Fikcyjne treści ułatwiają pracę, ale nie zwalniają z odpowiedzialności za konfigurację.

Kontrolowana droga na produkcję

01 Kopia

Pliki, baza i uploads; próba odtworzenia.

02 Staging

Osobna instalacja, HTTPS i kontrola dostępu.

03 Sprawdzenie

Treść, szablony, linki, klawiatura i układ.

04 Akceptacja

Właściciel zatwierdza wersję i ograniczenia.

05 Wdrożenie

Okno zmiany, kontrola po wdrożeniu i rollback.

Przenieś zgodne pliki motywu, bazę i media. Zmiana domeny wymaga narzędzia obsługującego serializację; ręczne zastępowanie tekstu w całym eksporcie może uszkodzić dane. Po migracji zapisz bezpośrednio odnośniki, otwórz pięć adresów oraz przejrzyj menu, stopkę, CTA i 404. Szczególną uwagę zwróć na linki od korzenia, jeżeli instalacja działa w podkatalogu. Wzorzec hero ma dynamiczny URL, lecz nie wszystkie przykłady są takie.

Kopia jest użyteczna dopiero po próbie odtworzenia. Zachowaj ją przed zmianą, odtwórz na odizolowanym środowisku i sprawdź kompletność. Jeśli produkcyjna baza zmieni się podczas pracy, nie zastępuj jej automatycznie starszą kopią stagingu. Uzgodnij sposób przeniesienia potrzebnych zmian oraz właściciela decyzji. Ograniczenie zakresu wdrożenia zmniejsza ryzyko utraty nowych treści.

09 / TESTY PAKIETU

Statyczna kontrola ma jasne granice

Mały walidator wychwytuje brak pliku i błędy struktury. Nie udaje parsera WordPress ani testu działającego serwera.

`validate_examples.py` sprawdza obecność dwunastu plików motywu, poprawność konfiguracji JSON i deklarację wersji 2. W szablonach, częściach, wzorcu oraz pięciu treściach kontroluje komentarze wp:, JSON atrybutów i zgodność odwołań do części. Sprawdza również brak przyszłego bloku projektów w aktywnym zestawie oraz brak formularza i mailto w Kontakcie. To konkretne zabezpieczenia kontraktu publikacji.

KOD W PRAKTYCE / PYTHON / Lista podstawowych plików

```
from pathlib import Path
import json, re

base = Path(__file__).resolve().parent
theme = base / "wp-demo-lab"
required = ["style.css", "functions.php", "theme.json",
            "templates/index.html", "templates/front-page.html",
            "templates/page.html", "templates/single.html",
            "templates/archive.html", "templates/404.html",
            "parts/header.html", "parts/footer.html", "patterns/hero.php"]
for name in required:
    assert (theme / name).is_file(), name
```

PLIK: `validate_examples.py`

Fragment pełnego pliku w `EXAMPLES`. Sprawdzona zgodność tekstu; uruchomienie WordPress/PHP niewykonane.

Wykonane uruchomienie zakończyło się powodzeniem: czternaście plików markup, 298 komentarzy blokowych i pięć treści stron. Wynik opisuje tę paczkę, a nie aktywny WordPress. Obecność `main` i `skip link` sprawdzono jako tekst. Interakcja, walidacja zapisu Gutenberg, odpowiedzi HTTP, błędy PHP i dostępność menu wymagają osobnego środowiska. Brak PHP oraz Docker został zapisany w `TEST_STATUS`, aby nie pomylić go z pomyślnym testem.

Po każdej zmianie pakietu możesz powtórzyć tę kontrolę. Jeżeli test przejdzie, wykonaj adekwatne scenariusze na stagingu; jeżeli nie przejdzie, najpierw napraw konkretną strukturę. Nie rozszerzaj deklaracji sukcesu poza sprawdzany zakres. W protokole zapisuj polecenie, rezultat i ograniczenie zamiast samego słowa `PASS`.

09 / ŹRÓDŁO PRAWDY

Synchronizacja plików i Site Editor

Zapis w Edytorze witryny może zasłonić plik motywu. Aktualizacja repozytorium nie zawsze zmienia to, co użytkownik widzi.

Założmy, że edytujesz `page.html` i dodajesz odstęp, lecz widok pozostaje taki sam. Sprawdź, czy szablon ma indywidualne dostosowanie w bazie. Podobnie działa zapisana część nagłówkowa i globalne style. Nie ma potrzeby dopisywać coraz silniejszych selektorów, zanim ustalisz źródło renderowanej wersji. Problem może dotyczyć warstwy danych, a nie skuteczności reguły CSS. Po ustaleniu źródła dopiero oceniaj pamięć podręczną.

Wybierz proces zespołu. Dla tego ćwiczenia pliki motywu są domyślną wersją projektową, a edytor służy próbom i świadomym zmianom. Zanim usuniesz nadpisanie, zachowaj kopię i porównaj jego treść z repozytorium. Zaakceptowane modyfikacje przenieś do plików lub jawnie pozostaw jako zarządzany rekord bazy. Dopiero wtedy wróć do domyślnego szablonu na stagingu i sprawdź wynik. Nie resetuj wszystkich dostosowań bez przeglądu.

Treści stron nie powinny być przenoszone do szablonu tylko po to, by znalazły się w Git. Właściciel ma nadal edytować ofertę w swoim rekordzie. Można zachować eksport demonstracyjnych treści jako punkt startowy, lecz po redakcji żywa baza pozostaje osobną warstwą. Podobnie nawigacja blokowa może otrzymać rekord i odwołanie zapisane przez edytor. Jej zawartość oraz adresy wymagają kontroli po migracji.

Przed wydaniem zapisz, które szablony mają nadpisanie i dlaczego. Uzupełnij instrukcję odtworzenia oraz wersję motywu. Kolejna osoba powinna wiedzieć, czy zmienia plik, rekord czy oba źródła w uzgodnionym procesie. To praktyczna część utrzymania witryny: mniej zgadywania, mniejsza liczba ukrytych zależności i łatwiejszy powrót do ostatniej zatwierdzonej wersji.

Odbiór wymaga demonstracji właściciela

Strona jest gotowa do przekazania dopiero wtedy, gdy można wykonać uzgodnione zadania oraz odtworzyć jej konfigurację.

Poproś właściciela o zmianę akapitu na stronie Usługi, zapis szkicu lub aktualizacji oraz obejrzenie podglądu. Następnie niech sprawdzi przejście z Home do Realizacji i Kontakt. Ta próba pokazuje, czy system jest zrozumiały i czy instrukcja odpowiada rzeczywistemu panelowi. Nie wykonuj wszystkich kroków za osobę odbierającą, bo wtedy nie dowiesz się, czy może później samodzielnie prowadzić witrynę.

PROMPT DO CODEXA / Raport odbioru bez wymyślonych wyników

Wklej do Codexa razem ze wskazanymi danymi.

Cel: przygotuj raport odbioru pięciostronicowej witryny WP DEMO LAB. Dane: zatwierdzony brief, STAGING_CHECKLIST.md, faktyczne wyniki testów i lista ograniczeń. Wynik: wymaganie, dowód, status OK/BŁĄD/NIE SPRAWDZONO, właściciel poprawki i decyzja wdrożeniowa. Brak dowodu oznacza NIE SPRAWDZONO. Uwzględnij próbę edycji przez właściciela, kopię i odtworzenie, menu, Kontakt oraz pięć URL. Nie dopisuj wyników PageSpeed ani aktywacji motywu.

KONTROLA: Porównaj raport z zapisami testera. Wymagania blokujące muszą mieć wykonany dowód; człowiek zatwierdza decyzję o publikacji.

W protokole oddziel błędy blokujące od zmian redakcyjnych. Niedziałający link, brak dostępu klawiaturą lub utrata treści wymagają naprawy przed wdrożeniem. Preferencja dotycząca długości nagłówka może być osobną decyzją, o ile komunikat pozostaje poprawny. Uzgodnione ograniczenia, takie jak kontakt bez wysyłki, nie są ukrytym błędem; muszą być jednak jasno widoczne dla odbiorcy i zapisane w przekazaniu.

Codex może przygotować formularz raportu oraz porównać wymagania z dowodami. Nie powinien zaznaczać wykonania zadania, którego nie potwierdza log lub obserwacja. Szczególnie odróżnij statyczne kontrole pakietu od testu WordPress i produkcyjnego wdrożenia. Taki podział pozwala uczciwie zatwierdzić etap oraz zaplanować brakujące sprawdzenia, bez tworzenia fałszywego poczucia ukończenia.

10 / PRZEKAZANIE I ROLLBACK

Po wdrożeniu sprawdź tę samą ścieżkę

Kontrola po publikacji obejmuje realne adresy i konfigurację. Udany staging nie zastępuje krótkiej próby w środowisku docelowym.

Przed zmianą zapisz wersję produkcyjną i uzgodnij okno wdrożenia. Przenieś zaakceptowane pliki i potrzebne dane, uwzględniając ewentualne nowe treści powstałe na produkcji. Po wdrożeniu otwórz pięć stron, sprawdź menu, CTA, HTTPS i 404. Zobacz logi oraz typowy widok mobilny. Wynik zapisz z datą i wersją, aby zgłoszenie późniejszego problemu miało punkt odniesienia.

Odpowiedzialne przekazanie

01 Wersja

Zatwierdzony zestaw plików, treści i ustawień.

02 Publikacja

Świadome wdrożenie w uzgodnionym oknie.

03 Kontrola

Pięć URL, menu, HTTPS, 404 i logi.

04 Reakcja

Naprawa lub zgodne przywrócenie kopii.

05 Utrzymanie

Instrukcja, właściciel aktualizacji i dalszy zakres.

Rollback powinien przywracać zgodny zestaw plików i danych. Sam starszy CSS nie naprawi błędnego rekordu nawigacji, a sama baza nie odtworzy brakującego zdjęcia. Plan wymaga właściciela decyzji, lokalizacji kopii i sprawdzonego sposobu odtworzenia. Po przywróceniu ponownie wykonaj podstawowe przejścia. W tej publikacji dostarczamy instrukcję takiego procesu, bez deklaracji, że rzeczywiste wdrożenie lub rollback wykonano.

Przełącz README, protokół, wersję motywu oraz informacje o nadpisaniach Edytora witryny. Ustal, kto utrzymuje aktualizacje i jak zgłaszać poprawki. Zachowaj rozdzielenie: tom 01 dostarcza motyw i pięć stron; tom 02 doda zarządzane rekordy realizacji; tom 03 wprowadzi kontrolowany szkic z formularza. Nie trzeba uruchamiać przyszłych funkcji, aby zaakceptować obecny, jasno ograniczony etap.

Dowody i ograniczenia pierwszego tomu

Wynik techniczny powinien być tak samo precyzyjny jak brief. Można dostarczyć kompletny przykład i uczciwie nazwać niewykonane testy.

Pakiet zawiera cały motyw wp-demo-lab, konfigurację lokalną, pięć treści, README, checkliście stagingu i walidator. Python odczytał JSON oraz sprawdził strukturę komentarzy bloków, wymagane pliki i odwołania części. Obliczył kontrast zadanych par. To wykonane czynności. Nie ma podstaw, by ogłosić pomyślną aktywację, php -l, test Gutenberg, test klawiatury lub wdrożenie: PHP i Docker nie były dostępne podczas przygotowania publikacji.

KOD W PRAKTYCE / PYTHON / Stosunek luminancji w walidatorze

```
def contrast(a, b):  
    values = sorted([luminance(a), luminance(b)])  
    return (values[1] + .05) / (values[0] + .05)
```

PLIK: validate_examples.py

Fragment pełnego pliku w EXAMPLES. Sprawdzona zgodność tekstu; uruchomienie WordPress/PHP niewykonane.

Fragment funkcji pokazuje rzeczywistą metodę porównania luminancji; pełny skrypt zawiera także przeliczenie sRGB. Wynik 2,97:1 dla mosiądzu i papieru oznacza brak odpowiedniego kontrastu dla zwykłego tekstu. Nie zaokrąglamy go do spełnienia progu. Las i atrament tworzą bazę tekstu, ale trzeba jeszcze sprawdzić wszystkie użyte stany i ewentualne obrazy. Pojedyncze obliczenie nie daje certyfikatu dostępności.

Następną czynnością czytelnika jest uruchomienie lokalnego WordPress i wykonanie checkliście. Zachowaj dowody oraz uzupełnij wynik. Dopiero po tej próbie możesz zatwierdzić wdrożenie we własnym środowisku. Kolejne tomy rozwijają tę samą demonstrację, dlatego pozostaw zgodne nazwy stron, motywu i trzech realizacji.

ŹRÓDŁA / STAN 13.10.2026

Dokumentacja i pliki do dalszej pracy

Adresy prowadzą do oficjalnej dokumentacji WordPress i W3C. Sprawdzaj wymagania w kontekście wersji używanej podczas własnego testu.

Theme Structure – Theme Handbook

<https://developer.wordpress.org/themes/core-concepts/theme-structure/>

Theme.json Version 2 Reference – Block Editor Handbook

<https://developer.wordpress.org/block-editor/reference-guides/theme-json-reference/theme-json-v2/>

Templates – Theme Handbook

<https://developer.wordpress.org/themes/templates/templates/>

Patterns – Theme Handbook

<https://developer.wordpress.org/themes/patterns/>

Global Settings and Styles – Theme Handbook

<https://developer.wordpress.org/themes/core-concepts/global-settings-and-styles/>

@wordpress/env – Block Editor Handbook

<https://developer.wordpress.org/block-editor/reference-guides/packages/packages-env/>

Understanding Success Criterion 1.4.3: Contrast (Minimum)

<https://www.w3.org/WAI/WCAG22/Understanding/contrast-minimum.html>

Understanding Success Criterion 2.4.7: Focus Visible

<https://www.w3.org/WAI/WCAG22/Understanding/focus-visible.html>

Źródła są oficjalnymi punktami odniesienia dla decyzji opisanych w tomie. Sprawdzono dokumentację konfiguracji, struktury, wzorców i środowiska oraz wybrane wymagania dostępności. Przykłady i teksty WP DEMO LAB stanowią samodzielny materiał demonstracyjny, nie kopię poradnika z jednej witryny. Interfejs panelu oraz zachowanie narzędzi mogą się zmieniać; zapisz wersję faktycznie używaną do ćwiczenia.

Pełne pliki znajdziesz w EXAMPLES, a wynik wykonanych kontroli w TEST_STATUS.md. Polecenia uruchomienia i lista stagingu opisują zadania czytelnika. Nie przedstawiają niewykonanych prób jako rezultatów. Minimum funkcji przykładu nie zastępuje wyboru aktualnie wspieranych wydań WordPress i PHP do własnego środowiska. Przy dalszej rozbudowie wracaj do dokumentacji właściwej dla konkretnej wersji.