

Checklista WCAG 2.2 – Poziomy A, AA, AAA

Potrzebujesz pomocy z wprowadzeniem zmian? - Skontaktuj się z nami na mk-web.pl

Dostępność cyfrowa staje się dziś jednym z kluczowych elementów profesjonalnych stron internetowych. Nowelizacja standardów WCAG 2.2 (Web Content Accessibility Guidelines) to istotny krok w stronę jeszcze większej dostępności serwisów internetowych – zarówno dla osób z niepełnosprawnościami, jak i użytkowników korzystających z technologii asystujących.

Poniżej znajdziesz rozpisane wszystkie kryteria WCAG 2.2, 2.1 oraz 2.0, z podziałem na poziomy, oraz instrukcją ich wdrożenia

Poziom A

Zasada 1: Postrzegalność

1.1.1 Treść nietekstowa

Zapewnij tekst alternatywny dla wszystkich treści nietekstowych, takich jak obrazy, ikony, wykresy, przyciski z grafiką. W praktyce dodaj atrybut „alt” do elementów `<img>`, a dla ikon lub grafik w tle – opis w treści lub za pomocą ARIA (np. `aria-label`).

1.2.1 Tylko audio i tylko wideo (nagranie)

Jeśli publikujesz nagranie audio lub wideo bez dźwięku, musisz dostarczyć alternatywę tekstową. Dla audio – transkrypcję, dla wideo – opis treści widocznej na ekranie.

1.2.2 Napisy rozszerzone (nagranie)

Dla nagrań wideo z dźwiękiem dodaj napisy zsynchronizowane z mową. Napisy powinny zawierać dialogi, ale też dźwięki otoczenia istotne dla zrozumienia treści.

1.2.3 Audiodeskrypcja lub alternatywa multimedialna

Zapewnij audiodeskrypcję (dodatkowy komentarz do akcji na ekranie) lub szczegółowy opis treści wideo w formie tekstowej.

1.3.1 Informacja i relacje

Używaj semantycznych elementów HTML, np. `<h1>`–`<h6>` dla nagłówków, `<ul>`, `<ol>`, `<li>` dla list, `<table>` dla tabel, `<label>` dla pól formularzy. Nie opieraj się wyłącznie na stylach wizualnych do przedstawiania relacji między elementami.

1.3.2 Kolejność odczytu

Zadbaj o to, by kolejność odczytu zawartości w kodzie HTML była logiczna i odpowiadała tej widocznej na ekranie. Nie używaj CSS do zmieniania kolejności elementów w sposób, który wprowadza w błąd osoby korzystające z czytników ekranu.

1.3.3 Właściwości sensoryczne

Nie polegaj wyłącznie na kolorze, kształcie, lokalizacji lub wielkości, aby przekazać informacje. Jeśli np. w instrukcji mówisz „kliknij czerwony przycisk po lewej”, dodaj również opis słowny lub ikonę.

1.3.4 Orientacja

Treść powinna działać niezależnie od orientacji ekranu (pionowej lub poziomej), chyba że zmiana orientacji jest niezbędna (np. w aplikacji do skanowania dokumentów).

1.3.5 Identyfikacja celu wejścia

Pola formularza (np. imię, e-mail, numer telefonu) powinny być odpowiednio oznaczone, tak aby systemy automatycznego wypełniania (np. w przeglądarkach) mogły je rozpoznać. Używaj `autocomplete="given-name"`, `autocomplete="email"` itp.

1.4.1 Użycie koloru

Kolor nie może być jedynym sposobem przekazywania informacji. Jeśli np. błędy w formularzu oznaczasz kolorem czerwonym, dodaj także tekstowy komunikat „To pole jest wymagane”.

1.4.2 Kontrola dźwięku

Jeśli strona automatycznie odtwarza dźwięk trwający dłużej niż 3 sekundy, użytkownik musi mieć możliwość zatrzymania go, wyciszenia lub regulacji głośności niezależnie od systemowego audio.

1.4.10 Przełamywanie bloków (reflow)

Zadbaj, by po powiększeniu zawartości (do 400%) użytkownik nie musiał przewijać poziomo, a tylko pionowo. Używaj elastycznych jednostek (em, %, vw) i media queries do responsywności.

1.4.11 Kontrast elementów nietekstowych

Elementy graficzne interaktywne (np. ikony przycisków, checkboxy) muszą mieć kontrast co najmniej 3:1 względem tła. Sprawdź to narzędziem typu WebAIM Contrast Checker.

1.4.12 Odstępy tekstu

Treść musi pozostać czytelna i funkcjonalna po zwiększeniu odstępów między znakami, wierszami i akapitami. Przetestuj CSS z ustawieniami: `line-height: 1.5`, `letter-spacing: 0.12em`, `word-spacing: 0.16em`.

1.4.13 Interaktywne wskaźniki

Wskaźnik klawiatury (tzw. focus) musi być widoczny dla każdego elementu interaktywnego. Unikaj usuwania stylu `outline` w CSS lub zamień go na inny kontrastowy styl.

Zasada 2: Funkcjonalność

2.1.1 Klawiatura

Wszystkie funkcjonalności strony muszą być dostępne z użyciem samej klawiatury. Upewnij się, że można poruszać się między elementami za pomocą klawisza TAB i aktywować je ENTEREM lub SPACJĄ.

2.1.2 Brak pułapki klawiatury

Użytkownik nie może „utknąć” na żadnym elemencie sterowanym klawiaturą (np. popupie, modalu). Zadbaj o możliwość powrotu klawiszem ESC, TAB lub przez przycisk „zamknij”.

2.1.4 Skróty klawiszowe

Jeśli oferujesz skróty klawiszowe (np. Shift + L), zapewnij możliwość ich wyłączenia lub zmiany przez użytkownika, szczególnie jeśli działają na elementach tekstowych.

2.2.1 Regulacja czasu

Umożliw użytkownikowi wydłużenie czasu trwania sesji lub działania formularza, np. poprzez opcję „przedłuż sesję”, „zapisz i kontynuuj później”.

2.2.2 Pauza, zatrzymanie, ukrycie

Dla animacji, slajderów lub automatycznie przewijających się treści zapewnij możliwość ich zatrzymania, wyłączenia lub ukrycia.

2.3.1 Trzy błyski lub mniej

Treści wizualne nie mogą błyskać więcej niż 3 razy w ciągu jednej sekundy. Unikaj efektów błyskających, które mogą wywołać atak epilepsji.

2.4.1 Unikalne tytuły stron

Każda podstrona musi mieć unikalny, opisowy tytuł (w tagu `<title>`), który jasno informuje o jej zawartości.

2.4.2 Porządek fokusu

Poruszanie się klawiaturą (TAB) musi odbywać się w logicznej, intuicyjnej kolejności odpowiadającej wizualnemu układowi strony.

2.4.3 Cel linków (w kontekście)

Linki powinny być opisane w sposób, który pozwala użytkownikowi zrozumieć ich cel – jeśli nie z samego linku, to z jego otoczenia. Przykład: „Więcej informacji o polityce prywatności”.

2.4.4 Sposoby nawigacji

Strona powinna udostępniać co najmniej jeden alternatywny sposób poruszania się, np. menu, mapa strony, wyszukiwarka.

2.5.1 Gesty wskazujące

Jeśli funkcje opierają się na gestach przeciągania, dotyku lub ruchu (np. suwak), zapewnij możliwość wykonania tych samych działań innym sposobem, np. przyciskiem.

2.5.2 Zrezygnuj z przypadkowych aktywacji

Elementy interaktywne nie powinny aktywować się tylko przez dotyk – powinny wymagać potwierdzenia (np. kliknięcia).

2.5.3 Etykiety widoczne

Etykiety pól formularzy i przycisków powinny być zgodne z tym, co widzi użytkownik i co przekazuje czytnik ekranu. Nie powinno być rozbieżności między `aria-label`, `title` a widocznym tekstem.

2.5.4 Aktywacja ruchowa

Nie wymagaj wykonywania ruchów urządzeniem (potrząsania, obracania), chyba że oferujesz alternatywę w postaci przycisku lub gestu dotykowego.

2.5.7 Ruch przeciągania

Jeśli funkcja wymaga przeciągania (drag and drop), zapewnij alternatywę opartą na klikaniu, np. przycisk „przenieś tutaj”.

2.5.8 Minimalny rozmiar celu

Elementy interaktywne (np. przyciski, linki) powinny mieć co najmniej 24x24 piksele, chyba że są częścią tekstu lub mają wystarczającą przestrzeń wokół.

Jeśli chcesz, mogę teraz przygotować równie szczegółowe opisy dla poziomu **AA** i **AAA**. Czy mam kontynuować?

Zasada 3: Zrozumiałość

3.1.1 Język strony

Określ język dokumentu w znaczniku `<html>` za pomocą atrybutu `lang`, np. `<html lang="pl">`. Dzięki temu czytniki ekranu poprawnie interpretują wymowę tekstu.

3.2.1 Interakcje

Zmiana wartości pola (np. wybór opcji z rozwijanej listy) nie może automatycznie przekierowywać na inną stronę, chyba że użytkownik został o tym uprzedzony. Jeśli zmiana wartości powoduje przeładowanie strony, poinformuj użytkownika tekstowo lub daj przycisk „zatwierdź”.

3.2.2 Nawigacja

Elementy nawigacyjne (np. menu, stopka) powinny być ułożone w tej samej kolejności na wszystkich stronach serwisu. Nie zmieniaj kolejności, nazw ani położenia tych samych komponentów między podstronami.

3.3.1 Identyfikacja błędów

W przypadku błędów w formularzach poinformuj użytkownika, co poszło nie tak i które pole wymaga poprawy. Komunikaty powinny być umieszczone bezpośrednio przy polu oraz być czytelne dla czytników ekranu.

Przykład: Pole tekstowe z błędem powinno mieć dodatkową klasę `aria-invalid="true"` oraz tekstowy komunikat np. `<span class="error">To pole jest wymagane</span>`.

3.3.2 Etykiety lub instrukcje

Każde pole formularza musi mieć wyraźną etykietę (`<label>`) lub widoczną instrukcję objaśniającą, jak je wypełnić. Etykieta powinna być powiązana z polem za pomocą atrybutu `for` i `id`, np.:

```
<label for="email">Adres e-mail</label>
<input id="email" name="email" type="email">
```

Zasada 4: Solidność

4.1.1 Poprawna składnia

Kod strony (HTML, XML) musi być poprawny składniowo:

- wszystkie znaczniki powinny być poprawnie otwarte i zamknięte,
- atrybuty powinny mieć poprawną składnię (`nazwa="wartość"`),
- nie powinno być zduplikowanych `id`.

Aby to wdrożyć:

- użyj walidatora W3C: <https://validator.w3.org>,
- popraw wszystkie błędy i ostrzeżenia związane z HTML i ARIA.

4.1.2 Nazwa, rola, wartość

Każdy element interaktywny (np. przycisk, suwak, pole formularza) musi mieć zdefiniowaną:

- **nazwę** (`aria-label` lub `<label>`),
- **rolę** (`role="button"`, `role="dialog"` itd.),
- **wartość** – np. czy checkbox jest zaznaczony (`aria-checked="true"`).

Dotyczy to szczególnie komponentów niestandardowych budowanych w JavaScript. Jeśli używasz np. własnych dropdownów, przycisków lub kart – muszą one być „widoczne” dla czytelników ekranu dzięki odpowiednim atrybutom ARIA.

Poziom AA

Zasada 1: Postrzegalność

1.2.4 Napisy (na żywo)

Dla materiałów wideo transmitowanych na żywo (np. webinary, transmisje wydarzeń) należy zapewnić napisy na żywo. Można to zrobić na dwa sposoby:

- przez zatrudnienie osoby do transkrypcji (np. w aplikacji Zoom, MS Teams),
- przez włączenie automatycznego generowania napisów (np. YouTube Live, Google Meet, OBS z wtyczką).

Napisy powinny obejmować wszystkie istotne dźwięki, nie tylko dialogi (np. „[muzyka w tle]”, „[śmiech publiczności]”).

1.2.5 Audiodeskrypcja (nagranie)

Jeśli w nagraniu wideo występują ważne informacje wizualne (np. wykresy, akcje bez dźwięku), należy zapewnić audiodeskrypcję – czyli narrację opisującą obraz.

Wdrażanie:

- Można przygotować osobną ścieżkę audio z opisem wizualnym.
- Można stworzyć wersję filmu z narracją w tle (np. lektor opisuje: „mężczyzna pokazuje wykres”).

Jeśli nagranie zawiera tylko mowę (np. prezentacja bez animacji), audiodeskrypcja nie jest wymagana.

1.3.6 Identyfikacja celu (Nowość w WCAG 2.2)

Każdy element interaktywny (link, przycisk, ikona) musi mieć jasno określony cel, również w przypadku, gdy wyświetlany jest tylko graficzny symbol (np. ikona koszyka, lupa).

Wdrażanie:

- Dla ikon używaj `aria-label`, `aria-labelledby` lub tekstu dostępnego tylko dla czytników (`sr-only`).
 - Przykład: ikona koszyka powinna mieć `aria-label="Dodaj do koszyka"`.
 - Linki „czytaj więcej” powinny być doprecyzowane – np. „czytaj więcej o WCAG 2.2”.
-

1.4.3 Kontrast (minimum)

Tekst oraz obrazy tekstowe muszą mieć kontrast względem tła na poziomie:

- minimum **4.5:1** dla tekstu zwykłego,
- minimum **3:1** dla tekstu dużego (min. 18pt bold lub 24pt regular).

Wdrażanie:

- Sprawdź kontrasty przy pomocy narzędzi takich jak WebAIM Contrast Checker, Color Contrast Analyzer.
 - Unikaj jasnoszarego tekstu na białym tle i innych słabo widocznych kombinacji.
-

1.4.4 Zmiana rozmiaru tekstu

Użytkownik musi móc powiększyć tekst do 200% bez utraty zawartości, możliwości interakcji ani potrzeby przewijania poziomego.

Wdrażanie:

- Stosuj jednostki względne (`em`, `rem`, `%`) zamiast `px`.
 - Unikaj kontenerów o stałej wysokości.
 - Projektuj responsywnie (media queries, elastyczne siatki).
-

1.4.5 Obrazy tekstowe

Nie stosuj obrazów zawierających tekst, chyba że:

- są niezbędne ze względów technicznych (np. logo),
- są częścią treści nieprzewidzianej do edycji (np. skany dokumentów).

Wdrażanie:

- Tekst powinien być zawsze edytowalny i renderowany jako tekst HTML lub CSS.
- Jeśli musisz użyć obrazu z tekstem (np. baner promocyjny), dodaj odpowiedni `alt` zawierający ten tekst.

Zasada 2: Funkcjonalność

2.4.5 Wiele dróg

Użytkownik powinien mieć co najmniej dwa sposoby na dotarcie do dowolnej strony w serwisie.

Wdrażanie:

- Udostępnij główne menu nawigacyjne.
- Dodaj jedną z poniższych opcji:
 - wyszukiwarkę,
 - mapę strony (sitemap),
 - listę kategorii lub indeks treści.

Wyjątek: to kryterium nie dotyczy stron będących częścią procesu (np. strona „Zamówienie – krok 3”).

2.4.6 Nagłówki i etykiety

Nagłówki sekcji oraz etykiety pól formularzy muszą być zrozumiałe i opisowe – powinny jasno mówić, co zawiera dana sekcja lub czego oczekuje się od użytkownika.

Wdrażanie:

- Zamiast ogólnego nagłówka „Informacje” użyj „Dane kontaktowe klienta”.
 - Etykieta pola formularza powinna być jasna: zamiast „Tekst” użyj „Wpisz swoją wiadomość”.
-

2.4.7 Widoczność fokusu

Podczas poruszania się klawiaturą (klawiszem TAB), aktywny element (np. link, przycisk) musi być wyraźnie oznaczony graficznie.

Wdrażanie:

- Użyj stylu `outline`, `box-shadow` lub `border` dla aktywnego elementu.
 - Unikaj całkowitego usuwania obramowania przez `outline: none`.
 - Kontrast obramowania względem tła powinien być dobrze widoczny.
-

2.5.5 Wprowadzanie pomocnicze

Jeśli użytkownik może uzupełnić dane automatycznie (np. imię, adres, telefon), należy umożliwić to poprzez `autocomplete`.

Wdrażanie:

- Ustaw atrybuty `autocomplete` dla pól formularzy:
 - `autocomplete="given-name"` – imię,
 - `autocomplete="family-name"` – nazwisko,
 - `autocomplete="email"` – adres e-mail,
 - `autocomplete="tel"` – numer telefonu.

Dzięki temu przeglądarka może podpowiedzieć dane użytkownika.

2.5.6 Uporczywe polecenia

Użytkownik nie powinien być zmuszony do wielokrotnego wpisywania tych samych informacji w trakcie jednego procesu (np. rejestracja, zakup).

Wdrażanie:

- Pamiętaj dane użytkownika między krokami formularza (np. przez `sessionStorage`).
- Jeśli dane są powtarzane – wypełnij je automatycznie lub pozwól je skopiować.
- Unikaj sytuacji, w których użytkownik musi podać np. adres dwa razy w tym samym formularzu.

Zasada 3: Zrozumiałość

3.1.2 Język fragmentów

Jeśli na stronie znajdują się fragmenty tekstu w innym języku niż główny język dokumentu, muszą być one odpowiednio oznaczone.

Wdrażanie:

- W treści strony otocz tekst w innym języku znacznikiem `span` z atrybutem `lang`.
 - Przykład:

```
<p>To jest słowo <span lang="en">accessibility</span> w języku angielskim.</p>
```
 - Dzięki temu czytniki ekranu automatycznie zmieniają sposób czytania (np. wymowę).
-

3.2.3 Spójność nawigacji

Nawigacja, która występuje wielokrotnie w serwisie (np. menu główne, stopka), powinna mieć zawsze ten sam układ i kolejność elementów.

Wdrażanie:

- Na każdej stronie menu powinno zawierać te same pozycje w tej samej kolejności.
 - Unikaj sytuacji, w której np. „Kontakt” znajduje się raz po lewej stronie, a innym razem po prawej.
-

3.2.4 Spójność komponentów

Komponenty interfejsu, które mają tę samą funkcję, powinny wyglądać i działać spójnie w całym serwisie.

Wdrażanie:

- Jeśli używasz przycisku „Zapisz” w formularzach, powinien on zawsze wyglądać tak samo i znajdować się w podobnym miejscu.
 - Jeśli kliknięcie na miniaturkę produktu na jednej stronie otwiera popup, to na innej stronie nie powinno przenosić do innej podstrony – powinno działać tak samo.
-

3.3.3 Sugestie naprawy błędów

Jeśli użytkownik popełni błąd w formularzu, system powinien zasugerować, jak go poprawić.

Wdrażanie:

- Oprócz komunikatu typu „Pole wymagane” dodaj informację, jak poprawnie wypełnić pole – np. „Podaj adres e-mail w formacie imie@domena.pl”.
 - W przypadku błędów z hasłem – np. „Hasło musi zawierać co najmniej 8 znaków, cyfrę i wielką literę”.
-

3.3.7 Dostępna pomoc (*nowość w WCAG 2.2*)

Użytkownik powinien mieć łatwy dostęp do pomocy na stronach zawierających złożone interakcje (np. zakup, formularz kontaktowy, rejestracja).

Wdrażanie:

- Dodaj widoczny link do strony pomocy lub FAQ.
 - Zapewnij dane kontaktowe (adres e-mail, numer telefonu, formularz).
 - Możesz dodać ikonę pomocy z podpowiedzią lub czat.
-

3.3.8 Ułatwienia wprowadzania (*nowość w WCAG 2.2*)

Jeśli użytkownik musi podać dane w formularzu, należy ograniczyć ilość wymaganych pól i maksymalnie uprościć proces.

Wdrażanie:

- Proś tylko o niezbędne dane.
- Użyj pól typu select, radio, checkbox zamiast pól tekstowych tam, gdzie to możliwe.
- Podaj format danych (np. dd-mm-rrrr) i dodaj podpowiedzi.
- Umożliw wypełnianie danych automatycznie (np. `autocomplete`).

Zasada 4: Solidność

4.1.3 Status komunikatów

Dynamiczne komunikaty (np. informacje o błędach, zapisaniu danych, dodaniu do koszyka) muszą być dostępne dla technologii asystujących, takich jak czytniki ekranu.

Wdrażanie:

- Umieszczaj komunikaty w elementach HTML z atrybutem `aria-live`:
 - `aria-live="polite"` – dla komunikatów informacyjnych (np. „Zapisano dane”),
 - `aria-live="assertive"` – dla pilnych komunikatów (np. „Wystąpił błąd podczas zapisu”).
- Alternatywnie możesz użyć `role="status"` dla komunikatów, które mają być odczytane automatycznie.

Przykład:

```
<div aria-live="polite" class="komunikat">  
  Zapisano zmiany pomyślnie.  
</div>
```

Dodatkowe zalecenia:

- Unikaj ukrywania takich komunikatów tylko wizualnie (np. w modalu bez fokusu).

- Upewnij się, że komunikat pojawia się dynamicznie w DOM – najlepiej po zmianie treści lub przesłaniu formularza.

Poziom AAA

Zasada 1: Postrzegalność

1.2.6 Język migowy (nagranie)

Dla nagrań wideo należy udostępnić tłumaczenie na język migowy.

Wdrażanie:

- Dodaj do nagrania drugie okno (np. w rogu ekranu) z osobą tłumaczącą na polski język migowy (PJM).
 - Można też dodać alternatywną wersję wideo z tłumaczem.
-

1.2.7 Rozszerzona audiodeskrypcja (nagranie)

W przypadku gdy nie da się zmieścić audiodeskrypcji w standardowych przerwach dialogowych, należy stworzyć osobną wersję nagrania z dłuższymi pauzami, umożliwiającymi opis wizualnych zdarzeń.

1.2.8 Alternatywa multimedialna (tekstowa)

Dla nagrań audio-wideo (zawierających jednocześnie obraz i dźwięk) należy zapewnić **kompletną alternatywę tekstową** – czyli szczegółowy opis zawartości w postaci pliku tekstowego.

Wdrażanie:

- Przygotuj osobny dokument lub sekcję strony z opisem:
 - dialogów,
 - efektów dźwiękowych,
 - zawartości wizualnej (np. "Kobieta siedzi przy biurku i mówi do kamery").

- Umieść go tuż obok osadzonego filmu lub jako osobny link: „Pobierz wersję tekstową”.
-

1.2.9 Napisy (na żywo) – wszystkie treści

Oprócz wideo – również **transmisje audio na żywo** (np. podcast na żywo, internetowe radio) muszą mieć **napisy na żywo**.

Wdrażanie:

- Transkrybuj na bieżąco lub użyj automatycznego narzędzia do napisów na żywo (np. otwarte napisy przez aplikację wspierającą rozpoznawanie mowy).
-

1.4.6 Kontrast (rozszerzony)

Tekst musi mieć **jeszcze wyższy kontrast** niż na poziomie AA:

- **7:1** dla zwykłego tekstu,
- **4.5:1** dla dużego tekstu.

Wdrażanie:

- Przetestuj kolory za pomocą narzędzi (np. WebAIM).
 - Przykładowe kombinacje o wysokim kontraście: #000000 na #FFFFFF, #003366 na #FFFFFF.
-

1.4.7 Tło dźwiękowe

Jeśli publikujesz nagrania dźwiękowe (np. lektor czyta tekst), nie mogą zawierać **rozpraszającej muzyki w tle**.

Wdrażanie:

- Upewnij się, że mowa nie jest zagłuszana przez muzykę.
- Jeśli używasz tła – niech będzie ciche lub daj możliwość jego wyłączenia.

1.4.8 Wizualna prezentacja

Użytkownik powinien mieć możliwość **kontrolowania**:

- kolorów tekstu i tła,
- odstępów między wierszami, akapitami, literami.

Wdrażanie:

- Umożliw zmianę stylów (np. przez ustawienia dostępności lub przycisk „Tryb wysokiego kontrastu”).
- Stosuj CSS, który pozwala na nadpisywanie stylów (czyli nie blokuj stylów użytkownika).
- Utrzymuj dobre praktyki typografii: `line-height: 1.5`, `letter-spacing: 0.12em`, `word-spacing: 0.16em`.

Zasada 2: Funkcjonalność

2.1.3 Klawiatura (bez wyjątków)

Wszystkie funkcjonalności muszą być dostępne **wyłącznie z klawiatury**, bez wyjątku.

Wdrażanie:

- Elementy nie mogą wymagać myszy, gestów ani ekranu dotykowego.
- Nawigacja i obsługa powinny być możliwe przy użyciu klawiszy TAB, ENTER, ESC, SPACJA.
- Testuj elementy niestandardowe – np. karuzele, suwaki – pod kątem pełnej obsługi klawiaturą.

2.2.3 Brak limitów czasu

Użytkownik nie może być ograniczony czasem – jeśli sesja wygasa, powinien mieć możliwość jej odnowienia **bez utraty danych**.

Wdrażanie:

- Unikaj automatycznego wylogowywania lub przerywania procesu bez możliwości kontynuacji.
 - Daj opcję „przedłuż sesję” lub „zapisz i wróć później”.
-

2.2.4 Przerwy

Użytkownik powinien mieć kontrolę nad automatycznymi przerwami (np. wylogowaniem po bezczynności).

Wdrażanie:

- Pozwól ustawić czas przerwy lub daj informację, ile czasu pozostało.
 - Możesz dodać suwak ustawień (np. „Wyloguj po 15, 30, 60 minutach”).
-

2.2.5 Ponowne uwierzytelnianie bez utraty danych

Jeśli użytkownik musi się ponownie zalogować, dane wprowadzone wcześniej nie mogą zostać utracone.

Wdrażanie:

- Przechowuj dane formularzy tymczasowo po stronie klienta (`localStorage`, `sessionStorage`) lub serwera.
 - Po zalogowaniu przywróć użytkownika do miejsca, w którym był.
-

2.3.2 Trzy błyski (rozszerzenie)

Treści wizualne nie mogą błyskać więcej niż **3 razy w ciągu jednej sekundy**, bez względu na kolor, kontrast i tło.

Wdrażanie:

- Unikaj animacji z błyskami, miganiem, światłami stroboskopowymi.
- Jeśli błyski są konieczne – upewnij się, że częstotliwość nie przekracza 3/s.

2.4.8 Lokalizacja

Użytkownik powinien zawsze wiedzieć, **gdzie się znajduje** w obrębie witryny.

Wdrażanie:

- Dodaj ścieżkę dostępu (breadcrumb): np. „Strona główna > Kategorie > Elektronika > Laptopy”.
- Oznacz aktualnie aktywny element w menu (np. przez klasę `aria-current="page"` lub `class="active"`).

2.4.9 Cel linków (samodzielnie)

Linki muszą być zrozumiałe **samodzielnie**, bez kontekstu otaczającego tekstu.

Wdrażanie:

- Zamiast „Kliknij tutaj” napisz „Zobacz naszą ofertę laptopów”.
- Jeśli używasz przycisków z ikonami, dodaj `aria-label` opisujące dokładnie cel.

2.4.10 Tytuły sekcji

Każda sekcja treści powinna mieć **jasny, opisowy tytuł**.

Wdrażanie:

- Używaj nagłówków `h2`, `h3` itd. z opisową treścią: np. „Zalety recyklingu plastiku” zamiast „Informacje”.
- Nagłówki powinny odzwierciedlać strukturę logiczną strony.

2.5.9 Przeciągnij i upuść – alternatywa (*kryterium z fazy roboczej, opcjonalne*)

Dla funkcji „drag and drop” należy zapewnić alternatywny sposób wykonania tej samej akcji.

Wdrażanie:

- Jeśli użytkownik przeciąga element do innego miejsca – umożliw również przesłanie go przyciskiem lub wyborem z listy.

Zasada 3: Zrozumiałość

3.1.3 Nieczęsty język

Trudne, techniczne lub specjalistyczne słowa muszą być **wyjaśnione** w kontekście lub dostępne do szybkiego sprawdzenia.

Wdrażanie:

- Dodaj słowniczek trudniejszych pojęć (np. na końcu strony).
 - Podaj definicję bezpośrednio w tekście – np. „Usługa caching (tymczasowe przechowywanie danych)...”.
 - Możesz użyć dymków podpowiedzi lub linków do sekcji „Co to znaczy?”.
-

3.1.4 Skróty

Jeśli używasz skrótów (np. „WCAG”, „RWD”), zapewnij ich pełne rozwinięcie przy pierwszym wystąpieniu.

Wdrażanie:

- Na początku tekstu napisz np. „WCAG (Web Content Accessibility Guidelines)”.
- Dla czytników ekranu możesz też dodać atrybut `abbr title="pełna nazwa"`:

```
<abbr title="Cascading Style Sheets">CSS</abbr>
```

3.2.5 Zmiany na żądanie

Zmiany treści (np. ładowanie nowych sekcji, przejścia, aktualizacje) mogą zachodzić **tylko wtedy, gdy użytkownik wyraźnie o to poprosi**.

Wdrażanie:

- Unikaj automatycznego przewijania, zamykania popupów, przeładowywania stron bez działania użytkownika.
 - Przykład: zamiast ładowania kolejnej sekcji automatycznie, użyj przycisku „Załaduj więcej”.
-

3.3.4 Zapobieganie błędom (dane prawne, finansowe, osobiste)

Formularze zbierające **krytyczne dane** (np. podpisy umów, dane bankowe, dane osobowe) muszą:

1. mieć możliwość **przeglądu** danych przed zatwierdzeniem,
2. oferować możliwość **edycji**,
3. **potwierdzać** dane przed wysłaniem.

Wdrażanie:

- Dodaj ekran „Podsumowanie danych” z przyciskiem „Edytuj”.
- Pokaż komunikat „Czy na pewno chcesz przesłać dane?” z opcją cofnięcia.
- Daj informację: „Dane zostaną zapisane po kliknięciu Zatwierdź”.

Zasada 4: Solidność

4.1.1 Poprawna składnia (*już na poziomie A – przypomnienie*)

Kod HTML, XML i ARIA musi być poprawny, by technologie asystujące mogły go prawidłowo interpretować.

Wdrażanie:

- Używaj <https://validator.w3.org>, by znaleźć błędy.
 - Każdy znacznik powinien być poprawnie otwarty i zamknięty.
 - Nie duplikuj `id` i nie używaj niedozwolonych atrybutów.
-

4.1.2 Nazwa, rola, wartość (*również z poziomu A – przypomnienie*)

Elementy interfejsu muszą mieć określone:

- **nazwy** (np. `aria-label`),
 - **role** (np. `role="button"`),
 - **wartości** (np. `aria-checked="true"`).
-

4.1.3 Status komunikatów (*poziom AA – przypomnienie*)

Komunikaty (np. błędy, powodzenie, informacje o dodaniu do koszyka) muszą być dostępne dla czytników ekranu poprzez `aria-live`, `role="status"` itd.

Uwaga: poziom AAA nie wprowadza nowych kryteriów dla zasady 4

Na poziomie AAA **nie dodano żadnych nowych** kryteriów dla tej zasady – obowiązują jedynie te z poziomów A i AA, opisane powyżej.

Potrzebujesz pomocy z wprowadzeniem zmian? - Skontaktuj się z nami na mk-web.pl