

WCAG 2.1 lista kontrolna (A/AA)

Poniższa lista kontrolna jest listą najczęściej występujących błędów. Nie zawiera wszystkich heurystyk, dlatego też może nie realizować wszystkich scenariuszy użytkowników. Została opracowana przez uniwersytet Yale tak aby w prosty sposób, pozwolić zweryfikować większość rozwiązań webowych.

Wymagania WCAG na poziomie - A

1.1.1 Treść graficzna [2.0]

- ✓ Wszystkie grafiki muszą posiadać atrybut alt w języku polskim tłumaczącym co znajduje się na danej grafice
- ✓ Opis alt nie powinien być dłuższy niż 125 znaków
- ✓ Jeżeli dodano skomplikowaną grafikę, taką jak infografika wymagany jest opis pod nią w formie opisu (powiązanego ARIA) lub zwykłego tekstu.
- ✓ Jeżeli grafika lub ikona jest wykorzystywana jako button lub link opis powinien zawierać cel, który ma realizować.
- ✓ Jeśli obrazy są elementem designu takim jak tło, której celem jest wyłącznie warstwa estetyczna, nie informacyjna. Wtedy powinny posiadać pusty znacznik alt (`alt=""`) lub być deklarowane za pomocą stylów CSS.
- ✓ Elementy które linkują w to samo miejsce powinny być pogrupowane w sekcje, zamiast deklarować do każdego z nich osoby atrybut a href. Należy minimalizować liczbę linków do tego samego miejsca.

1.2.1 Materiały audio oraz wideo

- ✓ Dla materiałów audio powinna być stworzona audio deskrypcja, zawierająca dialogi oraz dźwięki słyszane w audio.
- ✓ Dla materiałów wideo (bez audio) powinna zostać stworzona audio deskrypcja opisująca wydarzenia, miejsca, osoby oraz obiekty pojawiające się w danym wideo.

1.2.3 Materiały audio oraz wideo

- ✓ Dla materiałów wideo (nie trwających na żywo) powinna zostać dostarczona transkrypcja lub audio deskrypcja.

1.3.1 Informacje i relacje pomiędzy obiektami

- ✓ W treści powinny być wykorzystywane znaczniki semantyczne takie jak: nagłówki, listy, numeracje, wyróżnienia tekstu itp.
- ✓ Strony z treścią powinny być logicznie podzielone za pomocą nagłówków HTML (H1, H2, H3 itd.).
- ✓ W rozwiązaniach webowych powinny zostać zastosowane znaczniki ARIA oraz etykiety do identyfikacji obiektów na stronie oraz do wskazywania powiązań.
- ✓ Tabele powinny być wykorzystywane tylko dla danych tabelarycznych, tabele nie zostały wykorzystane do dzielenia treści lub dokumentu.
- ✓ Uniknięto emulacji oraz linków za pomocą div, span itp. Użyto takich tagów jak <a> lub <button>.
- ✓ Nie użyto białych znaków (zbędnych spacji, białych liter, czy fontów o niewidocznym rozmiarze).

1.3.2 Odpowiednia kolejność elementów

- ✓ Wyłączenie stylu CSS powinno wyświetlać treść na stronie w odpowiedniej kolejności.

1.3.3 Odpowiednia kolejność elementów

- ✓ Animacje i elementy wizualne powinny posiadać opis. Zrozumienie danej treści czy komunikatu nie powinno się opierać wyłącznie na wrażeniach zmysłowych takich jak ruch, kształt, rozmiar, kolor, umiejscowienie czy dźwięk. Przykładem takiego działania będzie animacja CSS.

1.4.1 Wykorzystanie koloru

- ✓ Pola wymagane, jak i błędy, powinny poza kolorem posiadać komunikat w postaci treści, który pozwoli na zrozumienie komunikatu.
- ✓ Kolor nie powinien być jedynym czynnikiem pozwalającym zrozumienie treści np. zaznaczenie błędu na czerwono bez stosownego komentarza.
- ✓ Linki poza kolorem, powinny być oznaczone również w inny niebudzący wątpliwości sposób, najbardziej popularnym jest podkreślenie.

1.4.2 Kontrola odtwarzania dźwięku

- ✓ Pliki dźwiękowe nie powinny być uruchamiane automatycznie. Funkcja uruchomienia jak i zatrzymania pliku powinna być łatwo dostępna.

2.1.1 Klawiatura

- ✓ Każda funkcjonalność powinna być dostępna z poziomu klawiatury.
- ✓ Implementowane skróty klawiszowe nie mogą ingerować w działanie czytników stron oraz w standardowe działanie przeglądarki (tab, strzałki). Najlepiej ich unikać.
- ✓ Wszystkie komunikaty czy informacje powinny być dostępne za pomocą kliknięcia lub klawiatury. Akcje takie jak tooltips, onmouseover czyli akcje polegające wyłącznie na wskazaniach czy ruchach myszy są nieużyteczne.
- ✓ Elementy aktywne, za pomocą klawiatury powinny posiadać wyraźną obwódkę.
- ✓ Pola typu input powinny posiadać migającą pionową linię gdy są aktywne i możemy coś w nich wpisać.

2.1.2 Anulowanie focusa

- ✓ Jeżeli za pomocą klawiatury np. za pomocą strzałek możemy aktywować focus danego pola, to również za pomocą klawiatury powinniśmy mieć możliwość jego usunięcia.

2.1.4 Przypisywanie skrótów klawiszowych

- ✓ Jeżeli istnieją specjalne skróty klawiszowe wykorzystujące wyłącznie litery, numery lub symbole użytkownik musi mieć możliwość ich dezaktywacji lub edycji takich skrótów.

2.2.1 Limitowanie czasu

- ✓ Nie limitujmy za pomocą czasu możliwości wykonania danego zadania. Jeżeli to konieczne limit czasowy nie powinien być mniejszy niż 20 godzin oraz powinien umożliwiać jego wydłużenie.

2.2.2 Ruchome elementy

- ✓ Elementy w rozwiązaniu nie powinny automatycznie się poruszać, uciekać, scrollować włączając w to karuzele. Występowanie elementów ruchomych powinno występować razem z panelem sterowania, który umożliwi ich zatrzymanie.

2.3.1 Migające elementy

- ✓ Elementy na stronie nie powinny migać częściej niż raz na sekundę.

2.4.1 Możliwość pominięcia

- ✓ Powinniśmy umożliwić pominięcie powtarzających się elementów. Chyba, że ich obecność jest niezbędna dla odbiorcy lub procesu.

2.4.2 Tytuły podstron

- ✓ Każda podstrona powinna posiadać unikalny tytuł, opisujący ekran oraz zawierający się w tagu <title>.

2.4.3 Kolejność focusa

- ✓ Zadbaj aby Focus przełączał się w odpowiedniej kolejności niezbędnej do zapoznania się z treścią lub wykonania zadania np. wypełnienia formularza.
- ✓ Focus powinien zachowywać odpowiednią kolejność, nawet po edycji, wywołaniu menu czy dodatkowej akcji.
- ✓ Tabindex nie powinien być wyższy niż "0".

2.4.4 Cel linku

- ✓ Docelowe miejsce linku powinno być zgodne z tekstem znajdującym się na nim.
- ✓ Jeżeli tekst nie jest do końca w stanie oddać celu do którego kieruje, wtedy powinien znaleźć się obok niego stosowny komentarz, można wykorzystać do tego metodologię ARIA.

2.5.1 Gesty

- ✓ Nie wymagaj gestów wielopunktowych (np. szczypanie, przeciąganie) dla funkcjonalności, chyba że gest jest niezbędny dla funkcjonalności. Jeżeli zastosowano takie rozwiązanie, powinno ono mieć odpowiednik w formie klasycznego menu/buttonu aktywującego daną funkcję

2.5.2 Anulowanie wskaźnika

- ✓ Unikaj funkcji "onmousedown", czyli akcji wymagających najechania, przytrzymania jakiegoś buttona. Zamiast tego powinno się stosować "onclick"

2.5.3 Etykieta w nazwie

- ✓ Nazwa, funkcja oraz etykieta muszą być spójne względem siebie.

2.5.4 Aktywacja ruchem

- ✓ Funkcje lub działania powinny unikać aktywacji ruchem takich jak: potrząśnięcie, ściśnięcie czy przytrzymanie. Jeżeli zastosowano takie rozwiązanie, powinno ono mieć odpowiednik w formie klasycznego menu/buttonu aktywującego daną funkcję.

3.1.1 Język strony

- ✓ Każda podstrona powinna zawierać atrybut "lang", wskazujący na język danej podstrony.

3.2.1 Stałość strony - focus

- ✓ W momencie zmiany miejsca focusa, nie powinny następować zmiany w interfejsie np. uzupełniać pól, zaznaczać elementów, dokonywać wyboru opcji.

3.2.2 Stałość strony - input

- ✓ Uzupełnianie pola typu input nie powinno dokonywać zmian w formularzu, wysyłania danych, aktywowania automatycznego okienek czy dodatkowych menu. Interfejs powinien być przewidywalny i reagować przewidywalnie dla użytkownika.

3.3.1 Walidacja, błędy

- ✓ Błędy powinny być oznaczone za pomocą "aria-invalid".
- ✓ Pola wymagane powinny być oznaczone za pomocą "aria-required" oraz widoczną informacją w interfejsie.
- ✓ Błędy powinny być: łatwe do zauważenia, identyfikacji oraz poprawienia.

3.3.2 Labelle do instrukcji

- ✓ Tekst poprzedzający pole typu input powinien wyjaśniać jaka zawartość powinna znaleźć się w polu.
- ✓ Opis pola i wyjaśnienia powinny zawierać również oznaczenie "aria-describedby"
- ✓ Opisy pól powinny znajdować się nad polami. Wyjaśnienia nie mogą znajdować się w polu placeholder.

4.1.1 Poprawność kodu

- ✓ Zweryfikuj poprawność kodu, błędy mogą spowodować błędne działanie czytelników treści.

4.1.2 Nazwa, rola, wartość

- ✓ Unikaj tworzenia elementów, które istnieją już w języku html np. link zastosuj tag `<a>` zamiast zagnieżdżać tą funkcję.

Wymagania WCAG na poziomie - AA

1.2.5 Audio deskrypcja

- ✓ Audio deskrypcja powinna być nagrana w formie radiowej, powinna zawierać emocje, opisy a nawet dźwięki, które pozwolą lepiej zrozumieć treść.

1.3.4 Orientacja

- ✓ Niezależnie od orientacji urządzenia, horyzontalnej czy wertykalnej witryna powinna prezentować czytelną i dostosowaną treść. Jeżeli jednak rozwiązanie nie pozwala prezentować go w różnych orientacjach trzeba to jasno zaznaczyć i ostrzegać podczas zmiany orientacji.

1.3.5 Autouzupelnianie

- ✓ Jeżeli formularz wymaga danych identyfikacyjnych, które mogą zostać auto uzupełnione, powinniśmy umożliwić przeglądarce użytkownika na automatyczne wypełnienie pól za pomocą HTML.

1.4.3 Minimalny kontrast

- ✓ Minimalny kontrast pomiędzy tekstem a tłem powinien wynosić 4.5:1. Dla fontów o minimalnej wielkości 24px lub 19px + bold wynosi 3:1.

1.4.4 Zmiana rozmiaru tekstu

- ✓ Przy zmianie wielkości tekstu za pomocą przybliżenia przeglądarki lub zmiany rozdzielczości nie powinny być obserwowane utraty treści ani funkcjonalności rozwiązania.
- ✓ Fonty powinny być definiowane za pomocą em lub % zamiast w pikselach



1.4.5 Tekst na grafikach

- ✓ Tekst nie powinien znajdować się na grafikach, chyba, że dotyczy to logotypu.


1.4.10 Reflow

- ✓ Rozwiązanie powinno być dostosowane do zmniejszonej rozdzielczości takiej jak 320px szerokości. Bez scrollowania horyzontalnego (horyzontalne scrollowanie mogą posiadać mapy, tabele i zestawienia).


1.4.11 Kontrast dla elementów nietekstowych

-  Kontrast dla elementów nietekstowych interfacu powinien wynosić minimum 3:1.
-  Wszystkie elementy nieaktywne, zmieniające się pod wpływem nacisku czy naszego działania również powinny posiadać minimalny kontrast na poziomie 3:1.


1.4.12 Text spacing

-  Przestrzeń między liniami tekstu powinna być definiowana w em lub % zamiast px.

1.4.13 Wskazanie lub oznaczenie focusem

-  Po najechaniu na element lub wskazaniu focusem element powinien wyraźnie się zaznaczyć. Użytkownik powinien mieć możliwość łatwego usunięcia tego zaznaczenia za pomocą klawisza esc.

2.4.5 Identyfikacja miejsca

-  Istotą tego wymagania jest łatwość w identyfikacji miejsca, w którym się znajdujemy. Ustalenie na jakiej stronie się znajdujemy powinno być możliwe za pomocą 2 niepowiązanych ze sobą identyfikatorów. Do tego możemy wykorzystać: menu, breadcrumbs, kategorie, progress bar etc.

2.4.6 Nagłówki i labele

- ✓ Etykiety i nagłówki muszą być jednoznaczne oraz unikalne. Powinny być przede wszystkim zrozumiałe dla zwykłego użytkownika.

2.4.6 Focus widoczność

- ✓ Styl focusa musi być mocno rzucający się w oczy. Border nie mniejszy niż 2px, kontrast powyżej 4.5:1.

3.1.2 Język elementu

- ✓ Jeżeli dany element, modułu wymaga znajomości innego języka, powinien wtedy znajdować się obok niego atrybut "lang" określający język.
- ✓ Tworząc nowy element, widget zweryfikuj jaką nadać mu etykietę zgodną z wymaganiami ARIA
- ✓ Wykorzystuj komponenty ARIA tylko i wyłącznie w przypadku gdy elementy nie są gotowymi komponentami HTML, <button> nie potrzebuje dodatkowego znacznika by być zrozumiałym dla czytelników.

3.2.3 Spójność nawigacji

- ✓ Elementy posiadające tę samą funkcjonalność powinny wyglądać tożsamo, znajdować się w tym samym porządku oraz w tych samych miejscach.
- ✓ Menu powinno wyglądać tożsamo na wszystkich podstronach, znajdować się w tym samym porządku oraz w tych samych miejscach.

3.2.4 Spójność komponentów

- ✓ Komponenty posiadające tę samą funkcjonalność powinny wyglądać tożsamo w stosunku do pozostałych serwisów, znajdować się w tym samym porządku oraz w tych samych miejscach.

3.3.3 Błąd w polu input

- ✓ Jeżeli wypełnione pole input wywołuje błąd, serwis powinien wspomóc w poprawieniu błędów np. poprzez dodawanie masek w pola, słownik poprawiający błędy itp. proaktywność względem wypełnianych pól.

3.3.4 Błędy legalistyczne

- ✓ Jeżeli błąd może powodować skutki prawne, powinniśmy poinformować o tym użytkownika i poprosić o potwierdzenie zapoznania się z komunikatem. Ponadto system powinien mu umożliwić łatwą korektę po otrzymanym ostrzeżeniu.

4.1.3 Status wiadomości

- ✓ Każdy pojawiający się status lub komunikat powinien być jasno opisany za pomocą znaczników ARIA (live regions lub alerts).