Чекліст для оцінки **доступності вебсайтів**

Міністерство цифрової трансформації України

Дія

Швеція
Sverige

UNDP

У цьому документі — ключові пункти для перевірки вебсайту на доступність. Зокрема, деталізовані вимоги WCAG 2.1 (рівні A та AA) та практичні коментарі для тестування ресурсу або формування вимог до розробників.

Це гайд, який допоможе зробити сайт доступним і зручним для користувачів із різними потребами. Зокрема, для:

- людей із порушеннями зору, які користуються програмами зчитування з екрана (скринридерами);
- людей із порушеннями слуху, яким потрібен додатковий текстовий або візуальний контент;
- людей із порушеннями опорно-рухового апарату, які користуються лише клавіатурою або спеціальними пристроями замість мишки;
- людей із порушеннями мовлення чи інтелектуального розвитку, яким важливі проста структура й зрозуміла навігація;
- людей із комбінованими порушеннями.

Документ розробив Центр компетенцій із цифрової доступності при ДП «ДІЯ».

Термінологія

У вимогах цього чекліста слова вживаються в таких значеннях:

- **має (повинен)** — це обов'язкова умова, яку не можна ігнорувати;
- **рекомендується** — не обов'язкова, але бажана умова, яка покращить доступність і зручність сайту;
- **може**, **можна** — один із можливих способів реалізації;
- **виняток** — ситуація, в якій обов'язкову вимогу можна не застосовувати.

WCAG — настанови з вебдоступності (Web Content Accessibility Guidelines) версії 2.1. WCAG доступний англійською та українською мовами.

Важливий — контент, без якого суттєво зміняться зміст або функціональність сторінки. Його неможливо повноцінно замінити іншим способом.

Інтерактивний елемент — будь-який елемент сайту, з яким взаємодіє користувач: посилання, кнопка, чекбокс, поле вводу, спадний список тощо. Він може бути створений як стандартними HTML-тегами (<button>, <select>, тощо), так і розроблений окремо.

Маркер фокуса — візуальне виділення елемента, який зараз у фокусі користувача. Зазвичай це контрастна рамка навколо елемента. Налаштовується за допомогою CSS. Має важливе значення для вебдоступності.

Мультимедійний матеріал, обмежений в часі — аудіо- або відеоконтент, який має певну тривалість: ролики, подкасти, прямі ефіри тощо. Для доступності такий контент має супроводжуватися субтитрами або текстовою розшифровкою.

Скринридер — допоміжна програма, яку найчастіше використовують незрячі люди. Вона:

- розпізнає текст та елементи на сторінці й озвучує їх;

- повідомляє назви, ролі й значення інтерактивних елементів під час фокусування на них;
- змінює навігацію з клавіатури: наприклад, стрілка вниз переходить до наступного елемента, а не просто прокручує сторінку.

Сторінка (вебсторінка) — окрема частина сайту з власною адресою. Це може бути як самостійна сторінка, так і крок у певному процесі.

Доступний для скринридера — елемент, який:

- скринридер може знайти й правильно озвучити;
- можна використовувати, коли скринридер увімкнений.

Прихований від скринридера — елемент, який скринридер не бачить. Наприклад, через застосування CSS-стилю **display: none**, атрибута **aria-hidden="true"** чи атрибута **alt=""**.

Програмно визначений — елемент, роль якого прописана в коді. Наприклад, що це кнопка, заголовок чи список. Завдяки цьому спеціальні програми (ті ж скринридери) правильно ідентифікують та озвучують цей елемент.

Синхронізовані мультимедійні матеріали — аудіо або відео, у яких звук і зображення працюють разом. Наприклад, відео зі звуком або аудіо з візуальним супроводом. Виняток — коли мультимедійний матеріал є альтернативою для тексту, і це чітко вказано.

Супровідні написи — розширені субтитри або SDH для людей із порушеннями слуху. Вони передають не лише репліки, а й важливі звуки для розуміння медіавмісту: музику, сміх, стукіт у двері тощо.

Фокусування — переведення фокуса клавіатури на елемент, із яким можна взаємодіяти. Наприклад, на стандартний інтерактивний елемент HTML або елемент з атрибутом **tabindex="0"**.

Інші специфічні терміни можна переглянути в Настановах з доступності вебвмісту WCAG.

Зміст

- Вимоги щодо програмного визначення і структурування
- Вимоги щодо керування
- Вимоги щодо дизайну та зовнішнього вигляду
- Вимоги щодо зображень
- Вимоги щодо мультимедійних матеріалів
- Інші вимоги
- Вимоги до впровадження та приймання послуг
- Корисні посилання

🗃️ Вимоги щодо програмного визначення і структурування

1. Заголовок сторінки (пункт 2.4.2 WCAG):

1.1. Сторінка **має** мати заголовок, який описує її тему або призначення, встановлений через тег **<title>**.

1.2. Рекомендується будувати заголовок за патерном «конкретне/загальне», а не «загальне/конкретне» (тобто «Новини/Міністерство», а не «Міністерство/Новини»).

2. Заголовки та маркування (пункт 2.4.6 WCAG):

2.1. Заголовки сторінок **мають** описувати тему чи призначення сторінок.

2.2. Заголовки частин (фрагментів) сторінок **мають** описувати тему чи призначення частин (фрагментів) сторінок.

2.3. Маркування компонентів **мають** описувати тему чи призначення компонентів.

2.4. Під описом теми чи призначення у вищеперерахованих підпунктах слід розуміти лаконічне узагальнення вмісту, а не сам вміст.

3. Інформація та взаємозв'язки (пункт 1.3.1 WCAG):

3.1. Інформація, структура та взаємозв'язки **мають** бути визначені програмно (або доступні в тексті), зокрема:

3.1.1. Заголовки **мають** відповідати **всім** указаним вимогам:

- Заголовок **має** бути програмно (семантично) визначений, а не просто бути жирним текстом збільшеного розміру (тобто слід застосовувати теги **<h1> — <h5>** АБО поєднання атрибутів **role="heading" aria-level="N"**, де N — потрібний рівень).
- Заголовки на сторінці **мають** починатися з рівня 1 (**<h1>**), однак **рекомендується**, щоб на сторінці був лише один заголовок рівня 1 (**<h1>**).
- **Не має** бути пропусків рівнів у заголовків на сторінці (тобто, наприклад, після **<h2>** **не має** йти **<h5>**).
- Заголовки **мають** логічно відображати структуру вмісту сторінки, еквівалентні за змістом елементи **повинні** мати той самий рівень, а не різні.
- Заголовки **не мають** застосовуватися до абзаців тексту чи інших елементів, які не виконують функцію заголовка.

3.1.2. Списки **мають** відповідати **всім** указанимвимогам:

- Списки **мають** бути програмно (семантично) визначені за допомогою тегів **<ul>**, **<ol>** та **<li>**.
- Безпосередніми нащадками елементів **<ul>** і **<ol>** **мають** бути елементи **<li>**.
- Списки **мають** застосовуватися для досягнення відповідної мети, зокрема не варто застосовувати список, якщо в ньому лише один елемент.

3.1.3. Таблиці **мають** відповідати **всім** указаним вимогам:

- Таблиці **мають** бути програмно (семантично) визначені за допомогою тегів **<table>**, **<tr>**, **<td>** та інших спеціально призначених для цього тегів.
- Комірки таблиць, які виконують роль заголовка для рядків чи стовпців, **мають** бути визначені за допомогою тегу **<th>**.
- Таблиці **не мають** застосовуватися для цілей побудови макета, верстки, тощо (тобто для суто візуальних цілей).
- **Рекомендується** не об’єднувати комірки таблиць (скринридери не можуть коректно інтерпретувати їх).
- **Рекомендується** не вкладати таблицю всередину іншої таблиці (наприклад, в одну з комірок).

3.1.4. «Шапка» сторінки **має** бути програмно (семантично) визначена за допомогою тега **<header>** АБО ** **.

3.1.5. «Підвал» сторінки **має** бути програмно (семантично) визначений за допомогою тега **<footer>** АБО ** **.

3.1.6. Основний вміст сторінки **має** бути програмно (семантично) визначений за допомогою тега **<main>** АБО ** **.

3.1.7. Інші взаємозв’язки та структура **мають** передаватися за допомогою призначених для цього тегів і атрибутів (таких, як **<nav>**, **<fieldset>**, тощо).

4. Назва, роль, значення (пункт 4.1.2 WCAG):

4.1. Всі компоненти інтерфейсу користувача (елементи форм, посилання, кнопки, створені за допомогою скриптів компоненти, тощо), **повинні** мати **всі такі** характеристики:

- програмно визначена **назва**, наприклад, за допомогою:
    - елемента **<label>** і його атрибута **for** (для елементів, які це підтримують);

    - атрибута **aria-label** (обов'язково, якщо кнопка не має видимого тексту і реалізована у вигляді графіки, іконки, емоджі, тощо);
    - атрибута **aria-labelledby** та пов’язаного елемента;
- програмно визначена **роль**, наприклад:
    - завдяки тому, що елемент реалізований на базі нативного тега (**<button>**, **<input>**, **<textarea>** тощо);
    - за допомогою атрибута **role**, якщо використаний елемент (тег HTML) не має власної програмної ролі (наприклад, ** ** чи ****);
- програмно визначені **стани** та **властивості**, якщо елемент здатний мати стани чи властивості, виходячи з його функції, наприклад за допомогою:
    - нативних властивостей **selected**, **checked**, **disabled**;
    - таких ARIA-атрибутів, як **aria-expanded**, **aria-current**, **aria-selected**, тощо, залежно від специфіки елемента;
- програмно визначене **значення**, якщо елемент здатний мати значення виходячи з його функції**,** наприклад за допомогою:
    - нативного атрибута **value**;
    - таких ARIA-атрибутів, як **aria-valuenow**;
- сповіщення про зміни в цих елементах є доступними для агентів користувача, зокрема допоміжних технологій (скринридерів тощо).

5. Маркування або інструкції (пункт 3.3.2 WCAG):

5.1. Кожен елемент інтерфейсу, який потребує введення даних користувачем, **повинен** мати текстову мітку, яка:

- пояснює, що потрібно ввести;
- візуально видима;
- програмно прив’язана до елемента через тег **<label>** чи атрибут **aria-label**.

5.2. Якщо елемент інтерфейсу призначений для введення даних у певному форматі (наприклад, РНОКПП, УНЗР) чи має обмеження (наприклад, обмеження кількості символів, заборона на введення певних символів), користувачу **має** бути надана текстова інструкція, яка:

- інформує про такі вимоги чи обмеження;
- візуально видима;
- програмно прив’язана до елемента через атрибут **aria-describedby**.

6. Маркування в назві елемента (пункт 2.5.3 WCAG)

6.1. Якщо інтерактивний елемент має видиму текстову мітку, цей самий текст і цією самою мовою **має** бути присутній у програмно визначеній назві цього елемента (стояти на її початку), яку читають асистивні технології (скринридери тощо). Це потрібно для коректної роботи асистивних (допоміжних) технологій.

6.2. Програмно визначена назва елемента **може** бути довшою, ніж цей видимий текст, щоб надати користувачам скринридерів важливу інформацію (наприклад, містити уточнення про те, що посилання відкриється в новій вкладці чи вікні).

7. Призначення полів уведення (пункт 1.3.5 WCAG):

7.1. Призначення кожного поля введення, що збирає інформацію про користувача, **має** бути визначено програмно, якщо поле **відповідає цим** умовам:

- поле введення має призначення (значення атрибута **autocomplete**), визначене в розділі WCAG «Цілі введення для компонентів інтерфейсу користувача»;
- вміст реалізовано за допомогою технологій з підтримкою визначення очікуваного значення для вхідних даних форми (наприклад, автозаповнення або маски, за допомогою яких система розуміє тип даних, які потрібно ввести в поле — дата, пошта, телефон, тощо).

8. Узгоджене розпізнавання (пункт 3.2.4 WCAG):

8.1. Компоненти, які мають однакову функціональність у наборі сторінок, **мають** однаково програмно визначатися. Приклади:

- Якщо інтерактивний елемент шукає щось на вебсайті і цей елемент є на різних сторінках сайту, то в усіх таких елементів:
  - **має** збігатися його програмна роль (кнопка);
  - **має** збігатися (якщо немає вагомих причин для протилежного) його текстова мітка, а не в кожному разі використовуватися оригінальна, наприклад: «Пошук», «Почати пошук», «Шукати», «Знайти» тощо.
- Якщо в якості кнопки на вебсайті використовується іконка (наприклад, 🔎), то така кнопка-іконка **має** одну визначену функцію (наприклад, функцію пошуку), а не різні функції на різних сторінках (наприклад, функцію пошуку в одному місці та функцію масштабування в іншому місці).
- Якщо текстова мітка «Закрити» в однієї кнопки означає закриття діалогового вікна, а в іншої кнопки закриття заявки (тобто скасування чи видалення

заявки), то слід уточнювати текстову мітку так, щоб користувач чітко розумів, що саме закривається.

9. Значуща послідовність (пункт 1.3.2 WCAG)

9.1. Якщо для правильного розуміння вмісту потрібно, щоб користувач зчитував елементи вмісту в конкретному порядку (правильна послідовність читання), такий порядок **має** бути доступний асистивним технологіям (скринридерам тощо). Тобто людина, яка не бачить екран, має почути цей текст у тому ж логічному порядку, у якому його читала б людина з хорошим зором. Найпростіший спосіб це забезпечити — виставити елементи вмісту в потрібній послідовності в коді HTML.

10. Порядок переміщення фокуса (пункт 2.4.3 WCAG)

10.1. Якщо послідовність руху фокуса клавіатури впливає на сенс та функціональність, то він **має** переміщатися між елементами в такому порядку, який дає змогу зберегти сенс та функціональність.

11. Узгоджена навігація (пункт 3.2.3 WCAG)

11.1. Механізми навігації, які повторюються на кількох сторінках у наборі сторінок, **мають** щоразу відтворюватись (розташовуватись) в тому самому відносному порядку, якщо користувач самостійно не змінював їх розташування. Наприклад:

- «Шапка» в усіх сторінок **має** розташовуватися в тому самому місці (нагорі), а її складові (посилання, кнопки) **мають** розташовуватися в тому самому відносному порядку.
- Якщо сторінки мають такі елементи, як загальний заголовок сторінки чи так звані «хлібні крихти» («breadcrumbs»), то цим елементам треба відводити одне й те ж місце в структурі сторінки (наприклад, вони мають стояти завжди на її початку).
- До механізмів навігації також належать: «підвал» сторінки, бічна навігаційна панель, панель пагінації, посилання чи кнопки для переходу назад і вперед.

12. Пропускання блоків (пункт 2.4.1 WCAG)

12.1. У шапці вебсайту (або перед нею) одним з перших елементів **має** бути розміщене посилання чи кнопка (далі — елемент), для якого/якої **мають** виконуватися **всі** такіі твердження:

- елемент **повинен** мати текстову мітку «Перейти до основного вмісту» чи схожу за змістом, ця мітка **має** бути видимою візуально та доступною для скринридерів;
- при активації елемента фокус клавіатури **має** стати на перший елемент основного вмісту сторінки;
- **рекомендується** за заданими параметрами робити елемент невидимим (для цього **не має** застосовуватися CSS-стиль **display: none**), в такому разі він **має** ставати видимим в момент фокусування на ньому (і **може** знову ставати невидимим після втрати фокуса).

Приклад реалізації елемента за посиланням.

⌨️ Вимоги щодо керування

13. Клавіатура (пункт 2.1.1 WCAG)

13.1. Весь функціонал вебсайту **має** бути доступним для клавіатури, тобто не вимагати виняткове використання миші чи інших засобів позиціонування курсора, зокрема, для таких дій:

- фокусування на всіх інтерактивних елементах (кнопки, перемикачі, прапорці, випадаючі списки тощо);
- виклик подій активації інтерактивних елементів (натискання кнопок, перемикачів, прапорців, розгортання спадних списків, вибір опцій у спадних списках тощо);
- перегляду спливних підказок (tooltips тощо).

14. Повне керування з клавіатури (пункт 2.1.2 WCAG)

14.1. Якщо фокус клавіатури можна перемістити на компонент сторінки за допомогою інтерфейсу клавіатури, тоді зміщення фокусу з цього компонента **має** відбуватись за допомогою лише інтерфейсу клавіатури. Тобто користувач, який використовує лише клавіатуру, **не має** потрапляти в ситуації так званої «пастки фокуса» («focus trap»), коли за допомогою клавіатури без повного оновлення сторінки не можна припинити фокусування на певному елементі.

14.2. Якщо для зміни фокусу не можна обійтися лише незміненими клавішами зі стрілками чи табуляцією або іншими стандартними методами виходу, тоді користувачу **мають** надаватися поради про те, як перемістити фокус.

15. Клавіші для швидких дій (пункт 2.1.4 WCAG)

15.1. Якщо в певному вмісті комбінація клавіш реалізується лише за допомогою літер (великих чи малих), знаків пунктуації, цифр або символів, тоді **має** справджуватися **принаймні одне** з таких тверджень:

- доступний механізм вимкнення комбінації;
- доступний механізм перепризначення комбінації для використання одного або кількох символів клавіатури, які не можна друкувати (наприклад, Ctrl, Alt тощо);
- комбінація клавіш для компонента інтерфейсу користувача активна лише тоді, коли цей компонент має фокус.

16. Керування жестами (пункт 2.5.1 WCAG)

16.1. Якщо на сторінці наявний функціонал, що потребує жестів з множинними дотиками або дотиками з певною траєкторією, і якщо такі жести не є критично важливими, то він **має** бути доступний і за допомогою єдиного курсора (тобто курсора введення, який працює з однією точкою контакту з екраном, охоплюючи поодинокі натискання та клацання, подвійні натискання та клацання, тривалі натискання та жести з певною траєкторією).

17. Скасування дій курсором (пункт 2.5.2 WCAG)

17.1. Для функціональних можливостей, якими можна керувати за допомогою єдиного курсора, **має** виконуватися **принаймні одне** з таких тверджень:

- Подія «down-event» курсора не використовується для виконання жодної частини функції (тобто в момент, коли користувач тільки натиснув, але ще не відпустив кнопку або палець («mousedown», «pointerdown», «touchstar» тощо), **не має** починатися чи завершуватися виконання будь-яких операцій).
- Завершення функції відбувається під час відпускання курсора, також доступний механізм для припинення функції до завершення або скасування функції після завершення.
- Відпускання курсора скасовує будь-який результат попереднього натискання.
- Завершення функції під час натискання є важливим.

18. Вміст під час наведення курсора чи встановлення фокуса (пункт 1.4.13 WCAG)

18.1. Якщо під час наведення курсора або фокусування за допомогою клавіатури стає видимим додатковий вміст, а під час їх відпускання він зникає, тоді **мають** справджуватися **всі** такі твердження:

- Можливість відхилити — **має** бути доступний механізм для відхилення додаткового вмісту без переміщення курсора або фокуса клавіатури, за **винятком** випадків, коли додатковий вміст:
    - або повідомляє про помилку введення;
    - або не приховує інший вміст;
    - або не замінює інший вміст.

  Наприклад, додатковий вміст **може** бути відхилений натисканням клавіші «Esc».

- Чутливість до рухів курсора — якщо при наведенні курсора на певний елемент з'являється додатковий вміст і курсор переміщається над цим додатковим вмістом, то він **не має** зникати.
- Тривалість — додатковий вміст **має** залишатися видимим доти, доки:
    - або не буде прибрано курсор або фокусування;
    - або користувач його не закриє;
    - або інформація не втратить валідність.

  Тобто автоприховування такого вмісту за певним таймером є помилкою.

19. Керування рухом (пункт 2.5.4 WCAG)

19.1. Якщо є функціонал, який керується за допомогою руху пристрою (струшування, нахил, обертання, тощо) або руху користувача (махання рукою перед камерою, тощо), то **мають** виконуватися обидві вимоги:

- функціонал **має** піддаватися керуванню за допомогою елементів користувацького інтерфейсу (кнопок, перемикачів тощо);
- **має** бути доступний спосіб вимкнути реакцію на рух, щоб запобігти випадковому спрацюванню.

Винятки, коли вищенаведені вимоги можна не виконувати:

- рух використовується для керування функціональними можливостями через інтерфейс, який підтримує доступність;
- рух є важливим для функціонування, а його вимкнення нівелює дію (наприклад, застосунок для здоров'я чи спорту).

20. Перебування у фокусі (пункт 3.2.1 WCAG)

20.1. Фокусування на будь-якому компоненті інтерфейсу користувача **не має** спричиняти жодних змін контексту.

21. Під час введення (пункт 3.2.2 WCAG)

21.1. Зміна налаштувань будь-якого компонента інтерфейсу користувача **не має** автоматично спричиняти жодних змін контексту, якщо користувач заздалегідь не поінформований про такі зміни внаслідок таких дій (попередження **має** бути як видиме візуально, так і доступне для скринридера). Під змінами налаштувань у цьому пункті маються на увазі, зокрема, такі дії:

- введення тексту в елементах **<input>**, **<textarea>** чи аналогічних;

- вибір опції в елементі **<select>** чи аналогічному;
- перемикання прапорців, радіокнопок чи подібних елементів;
- зміна значення (перетягування) повзунка (range) чи подібних елементів.

🎨 Вимоги щодо дизайну та зовнішнього вигляду

22. Контраст (мінімальні вимоги, пункт 1.4.3 WCAG)

22.1. Текст та зображення тексту **повинні** мати коефіцієнт контрастності не менш як **4,5:1** (порівнюється колір тексту з кольором фону, на якому текст розташований), за **винятком** таких випадків:

- Великий текст і зображення великого тексту **можуть** мати коефіцієнт контрастності **3:1**
- Вимоги щодо контрасту не висуваються до:
    - текстів і зображень тексту, які:
        - є частиною неактивного компонента інтерфейсу користувача (текст на **<button disabled>** чи плейсхолдер у **<input disabled>** тощо);
        - є оздобленням;
        - нікому не видно;
        - є частиною зображення, що містить істотний інший візуальний вміст;
    - текстів, які є частиною логотипу або назви бренду.

23. Контраст нетекстових елементів (пункт 1.4.11 WCAG)

23.1. Візуальне відображення наведених нижче елементів **повинно** мати коефіцієнт контрастності не менш як **3:1** із сусідніми кольорами:

- Компоненти інтерфейсу користувача — візуальна інформація, необхідна для ідентифікації компонентів інтерфейсу користувача та станів (до станів слід відносити і перебування у фокусі), за **винятком**:
    - неактивних компонентів;
    - випадків, коли зовнішній вигляд компонента визначається агентом користувача і не змінюється автором.
- Графічні об’єкти — частини графіки, необхідні для розуміння вмісту, за **винятком** випадків, коли конкретне подання графіки є важливим для інформації, що передається. Приклад такого винятку: фізичні чи температурні мапи, де плавна зміна кольорів передає плавну зміну висоти чи температури.

24. Застосування кольору (пункт 1.4.1 WCAG)

24.1. Колір **не має** використовуватися як єдиний візуальний засіб передачі інформації, вказівки на дію, спонукання до відповіді або розрізнення візуального елемента.

24.2. Зокрема, це означає, що всі гіперпосилання **мають** бути підкреслені або мати інше явне стилістичне відображення, яке відрізняє їх від звичайного тексту. Для гіперпосилань, які розташовуються прямо посеред тексту, **рекомендується** застосовувати саме нижнє підкреслення як традиційний засіб оформлення гіперпосилань. Допускається як підкреслення за замовчуванням, так і вдосконалене за допомогою CSS (приклад реалізації).

25. Зображення тексту (пункт 1.4.5 WCAG)

25.1. Для передавання нформації **має** використовуватися текст, а не зображення тексту. **Винятком** з цієї вимоги є такі випадки:

- зображення тексту можна візуально налаштувати під вимоги користувача (тобто встановити шрифт, розмір, колір і фон);
- конкретне подання тексту є важливим для інформації, що передається (наприклад, логотипи).

26. Видимий фокус (пункт 2.4.7 WCAG)

26.1. Всі елементи, які здатні отримувати фокус клавіатури, **мають** у момент отримання фокуса клавіатури отримувати візуально видимий маркер фокуса.

27. Зміна розміру тексту (пункт 1.4.4 WCAG)

27.1. Користувачу **має** бути забезпечена можливість змінювати розмір (масштаб) тексту без допоміжних технологій, збільшуючи його до 200 відсотків включно, так, щоб при цьому не втрачався вміст (інформація) або функціональність. Сайт має коректно масштабуватися при використанні стандартних клавіш Ctrl та +. **Винятком** з цієї вимоги є супровідні написи та зображення тексту.

28. Прокрутка в кількох напрямах (пункт 1.4.10 WCAG)

28.1. Вміст **має** бути представлений без втрати вмісту (інформації) чи функціональності й без необхідності прокручування у двох вимірах для:

- прокрутки вмісту на ширину, еквівалентну **320** пікселів CSS;
- прокрутки вмісту на висоту, еквівалентну **256** пікселів CSS.

Ця вимога не поширюється (**виняток**) на частини вмісту, які потребують двовимірного макета для належного використання або збереження смислу, до яких можна віднести:

- зображення;
- карти;
- діаграми;
- відео;
- ігри;
- презентації;
- таблиці даних;
- інтерфейси, де необхідно тримати панелі інструментів у полі зору під час маніпулювання роботи з вмістом.

29. Орієнтація (пункт 1.3.4 WCAG)

29.1. Перегляд вмісту всіх сторінок та взаємодія користувача зі вмістом всіх сторінок **мають** бути доступні і в портретній (книжковій, вертикальній), і в альбомній (горизонтальній) орієнтаціях дисплея. **Винятком** з цієї вимоги є ситуації, коли конкретна орієнтація є важливою. Зміна орієнтації **не має** призводити до втрати вмісту (інформації) чи функціональності.

29.A. Втрата вмісту (інформації) чи функціональності

29.A.1. До проявів втрати вмісту (інформації) чи функціональності, про які йдеться в контексті пунктів **1.4.4**, **1.4.10**, **1.3.4** WCAG, слід відносити:

- зникнення (обрізання) тексту, як суто візуальне, так і заскриптоване;
- вихід тексту за край вікна чи контейнера;
- перетинання (накладання) з іншими текстами чи нетекстовими елементами, що істотно ускладнює розпізнавання чи читання тексту;
- зникнення інтерактивних елементів разом з відсутністю появи аналогічних елементів їм на заміну;
- перетинання (накладання) інтерактивних елементів, що ускладнює чи унеможливлює вибір курсором миші потрібного елемента.

29.A.2. Приклади прийомів, які можуть завадити такій втраті:

- використання responsive layout (Flexbox / Grid);
- вказувати ширини елементів не в **px**, а у **%**, **em**, **rem** тощо;
- дозволяти переноси рядків (**word-wrap**, **overflow-wrap**);
- не використовувати **overflow-x: hidden** та подібні техніки;
- не використовувати фіксоване обмеження висоти елемента, який містить текст.

30. Міжрядковий інтервал (пункт 1.4.12 WCAG):

30.1. Якщо вміст реалізований з використанням мов розмітки, які підтримують наведені нижче властивості стилю тексту, то вміст і функціональність **мають** зберігатися (не втрачатися) при встановленні користувачем всіх перерахованих нижче параметрів (за умови незмінності всіх інших властивостей стилю):

- висота рядка (міжрядковий інтервал) **принаймні** в **1,5** рази перевищує розмір шрифту;
- відстань між абзацами **принаймні** у **два** рази перевищує розмір шрифту;
- інтервал між літерами (відстеження) **принаймні** в **0,12** рази перевищує розмір шрифту;
- інтервал між словами **принаймні** в **0,16** рази перевищує розмір шрифту.

Приклади втрати вмісту чи функціональності:

- тексти зникають, обрізаються чи виходять за межі вікна чи своїх контейнерів;
- тексти перетинаються (накладаються одне на одного);
- інтерактивні елементи (кнопки та ін.) зникають, перетинаються, спотворюються тощо;
- відбуваються візуальні спотворення інтерфейсу (верстки), які ускладнюють сприйняття інтерфейсу та взаємодію з ним.

31. Пауза, зупинення, приховування (пункт 2.2.2 WCAG)

31.1. Для будь-якого переміщення, миготіння чи прокручування (далі — такий процес), яке (1) запускається автоматично, (2) триває більш як п’ять секунд і (3) представлене паралельно з іншим вмістом, **має** бути наявний механізм, за допомогою якого користувач може призупинити, зупинити або приховати такий процес, за **винятком** тих випадків, коли такий процес є частиною *важливої* діяльності.

31.2. Для будь-якого автоматичного оновлення інформації (далі — такий процес), яке (1) запускається автоматично та (2) представлене паралельно з іншим вмістом, **має** бути наявний механізм, за допомогою якого користувач може призупинити, зупинити або приховати такий процес або керувати частотою оновлення, за **винятком** тих випадків, коли такий процес є частиною *важливої* діяльності.

32. Обмеження на три чи менше спалахи (пункт 2.3.1 WCAG)

32.1. Сторінки **не мають** містити:

- нічого, що блимає чи спалахує більше ніж 3 рази протягом секунди;
- спалаху, нижчого за загальні порогові значення спалаху, та червоного спалаху.

🖼️ Вимоги щодо зображень

33. Нетекстовий вміст (пункт 1.1.1 WCAG)

33.1. Зображенням, які не визначені як **винятки** в пункті 1.1.1 WCAG, **має** бути вказаний альтернативний текст в атрибуті **alt**, що належним чином описує їх.

33.2. **Рекомендується**, щоб довжина альтернативного тексту не перевищувала 150 символів. Від цієї рекомендації **можна** відступати, коли важливо надати повний, якісний опис зображеного.

🎞️ Вимоги щодо мультимедійних матеріалів

34. Лише аудіо та лише відеовміст у записі (пункт 1.2.1 WCAG)

34.1. Для мультимедійних матеріалів у записі, які містять лише аудіо та лише відео (за **винятком** випадків, коли аудіо або відео є альтернативою для тексту та чітко позначені як такі), **мають** виконуватися **всі** такі твердження:

- Для попередньо записаного аудіо **має** бути надана альтернативна версія у вигляді транскрипту з повним текстом діалогів, описом важливих звукових ефектів (наприклад, *«звук розбитого скла»*) та ідентифікацією того, хто говорить.
- Для попередньо записаного відео **має** бути надана альтернативна версія мультимедійного матеріалу (транскрипт), або аудіодоріжка, яка надає еквівалентну інформацію для виключно відеовмісту в записі.

35. Супровідні написи в записі (пункт 1.2.2 WCAG)

35.1. Аудіовміст у записі (попередньо записаний) у складі синхронізованих мультимедійних матеріалів (за **винятком** випадків, коли мультимедійні матеріали є альтернативою тексту і чітко позначені як такі) **повинен** мати супровідні написи (субтитри, які передають зміст усного діалогу та еквівалент аудіоінформації, яка не є

діалоговою, але необхідна для розуміння змісту програми, зокрема це звукові ефекти, музика, сміх, ідентифікація мовця та місцезнаходження).

36. Аудіоопис або альтернативні версії мультимедійних матеріалів у записі (пункт 1.2.3 WCAG)

36.1. Відео зі звуком обов'язково **повинне мати** текстовий опис усіх подій та діалогів, або додаткову аудіодоріжку, яка пояснює візуальні дії. Виняток — якщо саме відео вже є додатком, який просто дублює основний текст на сторінці.

37. Супровідні написи в режимі реального часу (пункт 1.2.4 WCAG)

37.1. Для будь-яких прямих трансляцій зі звуком (наприклад, вебінарів, онлайн-конференцій або стрімів) мають бути надані супровідні написи.

38. Аудіоопис у записі (пункт 1.2.5 WCAG)

38.1. Аудіоопис **має** бути наданий для всіх попередньо записаних відео зі звуком.

39. Аудіокерування (пункт 1.4.2 WCAG)

39.1. Якщо будь-яке аудіо (звук) на сторінці відтворюється автоматично протягом понад трьох секунд, то **має** бути доступний:

- або механізм для призупинення (pause) чи припинення (stop) аудіо;
- або механізм керування гучністю звуку незалежно від загального рівня гучності системи.

📦 Інші вимоги

40. Мова сторінки (пункт 3.1.1 WCAG):

40.1. Для сторінки мова вмісту **має** бути визначена програмно.

40.2. Для української мови налаштування **має** бути таким: **<html lang="uk">**.

41. Мова частин вмісту (пункт 3.1.2 WCAG):

41.1. Мова кожного уривку чи фрази у вмісті, якщо вона не збігається з основною мовою сторінки (вказаною в елементі **<html>**) **має** бути визначена програмно, за **винятком**:

- власних назв,
- технічних термінів,
- слів невизначеної мови,
- слів чи фраз, що є частиною говірки, використовуваної в тексті.

42. Сенсорні характеристики (пункт 1.3.3 WCAG)

42.1. Інструкції, надані для розуміння та використання вмісту, **не мають** покладатися лише на такі сенсорні характеристики компонентів, як форма, колір, розмір, візуальне розташування, орієнтація або звук. Тобто, наприклад, не можна робити вказівки на кшталт: «Щоб надіслати заявку, натисніть круглу зелену кнопку».

43.2. Призначення посилання в контексті (пункт 2.4.4 WCAG)

43.2. Призначення кожного посилання **має** визначатися лише з тексту посилання або з тексту посилання разом із програмно визначеним контекстом посилання, за **винятком** випадків, коли призначення посилання є неоднозначним для користувачів загалом (тобто коли факт наявності чи відсутності інвалідності не впливає на здатність зрозуміти, куди веде посилання). Приклад: детальніше ознайомитись з інформацією можна на сайті Мінцифри. Посилання зашите в текст «на сайті Мінцифри», з якого можна чітко визначити призначення посилання, — сайт Мінцифри.

44. Налаштування часу (пункт 2.2.1 WCAG)

44.1. Якщо використовуються ліміти часу (обмеження часу на виконання дій, таймери, тощо), то **має** виконуватися **принаймні одне** з таких тверджень:

- Користувач може вимкнути ліміт часу до того, як він з ним зіткнеться.
- Користувач може налаштувати ліміт часу, перш ніж він з ним зіткнеться, у широкому діапазоні, який **принаймні в 10 разів** перевищує тривалість налаштування за замовчуванням.
- Користувач отримує попередження до закінчення часу і має **принаймні 20 секунд**, щоб продовжити ліміт часу простою дією (наприклад, «натиснути пробіл»), користувач може в такий спосіб продовжити ліміт часу **щонайменше 10 разів**.
- Ліміт часу є обов'язковою частиною події в реальному часі (наприклад, аукціону), і альтернатива ліміту часу неможлива.
- Ліміт часу є *важливим* **винятком** (тобто відмова від нього докорінно змінить інформацію або функціональність вмісту, і цю інформацію та функціональність не можна забезпечити іншим способом, що характеризувався би відповідністю), і його подовження призведе до недійсності дії.
- Ліміт часу перевищує 20 годин.

45. Різні способи пошуку (пункт 2.4.5 WCAG)

45.1. **Має** бути забезпечена доступність (наявність) більш як одного способу знайти сторінку в наборі сторінок, за **винятком** випадків, коли сторінка є результатом процесу або його кроком. Приклади можливих способів знайти сторінки:

- посилання в шапці чи підвалі сайту;
- посилання в інших навігаційних панелях (контейнерах);
- посилання в основному вмісті інших сторінок;
- функціонал з пошуку сторінок (шляхом введення елементів їхньої назви, вмісту чи ключових слів);
- відображення списку пов'язаних сторінок у межах конкретної сторінки;
- мапа сайту (якщо це окрема сторінка, то вона **повинна** мати також більше одного способу бути знайденою).

46. Виявлення помилок (пункт 3.3.1 WCAG)

46.1. Якщо на інтерфейсі відображається помилка у відповідь на надану користувачем інформацію (введений текст, тощо), то **має** бути виконано **все** перераховане нижче:

- чітко ідентифіковано елемент, з яким пов’язана помилка (поле, в якому користувач ввів неправильні дані, тощо), така ідентифікація **має** бути здійснена обома способами:
    - візуально (наприклад, у поля виникає характерна обводка та/або поряд з полем виникає характерна іконка);
    - програмно (наприклад, у поля з’являються атрибути **aria-invalid="true"**).
- поряд з відповідним елементом **має** бути відображений текст, який описує суть помилки і спосіб її виправлення та пов’язаний з елементом програмно, наприклад, за допомогою атрибута **aria-describedby="{id-of-error-text}"**, де замість **{id-of-error-text}** треба підставити ідентифікатор цього тексту.

46.2. Якщо перше відображення чи оновлення вже відображених помилок пов’язується зі взаємодією користувача з елементом, який сам по собі не містить цих помилок (наприклад, кнопка «Надіслати форму»), то **рекомендується** на додачу до перерахованого вище відображати певне загальне оповіщення про наявні помилки поряд з цим елементом.

47. Підказування в разі помилок (пункт 3.3.3 WCAG)

47.1. Коли автоматично виявляється помилка введення і відомі можливі варіанти її виправлення, тоді відповідні пропозиції **мають** бути надані користувачу, якщо це не загрожує безпеці або призначенню вмісту.

48. Недопущення юридичних та фінансових помилок під час уведення даних (пункт 3.3.4 WCAG):

48.1. Для функціоналу сторінок, який:

- або призводить до юридичних зобов’язань чи фінансових операцій для користувача,
- або змінює чи видаляє контрольовані користувачем дані в системах зберігання даних,
- або надає відповіді на тестування користувача,

має справджуватися **принаймні одна** з таких характеристик:

- оборотність (наведені дані є оборотними);
- перевірка (введені користувачем дані перевіряються на наявність помилок і користувачеві надається можливість їх виправити);

- підтвердження (доступний механізм перегляду, підтвердження та виправлення інформації до завершення подання).

49. Синтаксис (пункт 4.1.1 WCAG)

49.1. Мови розмітки, якими викладається вміст на сторінках, **повинні** мати **всі** такі характеристики:

- елементи мають повні початкові та кінцеві теги, тобто:
  - наявний закривальний тег, якщо елемент вимагає його наявності;
  - наявні необхідні кутові дужки;
  - наявні лапки з відповідним значенням атрибута, лапки закриті коректно.
- елементи вкладені відповідно до їхніх специфікацій (наприклад, **<li>** вкладений в **<ul>**, **<ol>** тощо);
- елементи не містять повторюваних атрибутів;
- будь-які ідентифікатори є унікальними.

Винятком є випадки, коли специфікації конкретної мови розмітки чи елемента дозволяють протилежне.

50. Повідомлення про статус (пункт 4.1.3 WCAG)

50.1. У вмісті, реалізованому за допомогою мов розмітки, повідомлення про статус **мають** бути програмно визначені за допомогою ролі або властивостей, що забезпечить можливість представити їх користувачеві за допомогою допоміжних технологій без отримання фокуса. До повідомлень про статус належить, зокрема:

- оголошення про початок деякого тривалого процесу (завантаження файла на сервер, виконання запиту тощо);
- оголошення про завершення деякого тривалого процесу;
- оголошення про результат деякої значимої дії (видалення запису, анулювання заявки, додавання товару в кошик тощо).

☑️Вимоги до впровадження та приймання послуг

Порядок тестування має містити перевірку доступності всіх унікальних сторінок сайту на дотримання перерахованих вище вимог, а також:

1. Перевірка за допомогою браузерного розширення «Accessibility Insights for Web» (опція «FastPass»). Очікуваний результат:

1.1. Відсутність будь-яких помилок у вікні зі звітом, яке відкривається після активації опції «FastPass».

2. Перевірка доступності функціоналу сайту для клавіатури (тобто без використання миші). Очікуваний результат:

2.1. Весь функціонал сайту доступний для клавіатури, а не лише для миші чи інших засобів позиціонування курсора (крім дуже специфічних винятків).

2.2. Відсутні «пастки фокуса».

3. Перевірка за допомогою скринридера NVDA на ОС Windows. Очікуваний результат:

3.1. Порядок читання скринридером відповідає тому порядку, який передбачається для читання зрячим користувачем.

3.2. Скринридер коректно озвучує назви, ролі, значення (за наявності), стани (за наявності), властивості (за наявності) елементів інтерфейсу, на яких фокусується.

3.3. Вся функціональність сайту (крім дозволених винятків) доступна для скринридера, а в разі недоступності надана доступна альтернатива та відповідні інструкції, щоб незрячий користувач міг цим скористатися.

🔗Корисні посилання

Настанови з доступності вебвмісту (WCAG) 2.1

Приклади втілення деяких критеріїв WCAG

Перелік посилань на 50 пунктів WCAG 2.1, які мають рівень A або AA:

Рівень A: **1.1.1**, **1.2.1**, **1.2.2**, **1.2.3**, **1.3.1**, **1.3.2**, **1.3.3**, **1.4.1**, **1.4.2**, **2.1.1**, **2.1.2**, **2.1.4**, **2.2.1**, **2.2.2**, **2.3.1**, **2.4.1**, **2.4.2**, **2.4.3**, **2.4.4**, **2.5.1**, **2.5.2**, **2.5.3**, **2.5.4**, **3.1.1**, **3.2.1**, **3.2.2**, **3.3.1**, **3.3.2**, **4.1.1**, **4.1.2**

Рівень AA: **1.2.4**, **1.2.5**, **1.3.4**, **1.3.5**, **1.4.3**, **1.4.4**, **1.4.5**, **1.4.10**, **1.4.11**, **1.4.12**, **1.4.13**, **2.4.5**, **2.4.6**, **2.4.7**, **3.1.2**, **3.2.3**, **3.2.4**, **3.3.3**, **3.3.4**, **4.1.3**