

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Острозька академія»
Навчально-науковий інститут інформаційних технологій та бізнесу
Кафедра інформаційних технологій та аналітики даних

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня бакалавра

на тему: **«Імплементация стандарту WCAG (Web Accessibility) для системи управління бібліотекою з онлайн каталогом та резервуванням книг»**

Виконав: студент 4 курсу, групи КН-42
першого (бакалаврського) рівня вищої освіти
спеціальності 122 Комп'ютерні науки
освітньо-професійної програми «Комп'ютерні науки»
Фасоля Валентин Олегович

Керівник: викладач кафедри інформаційних технологій та
аналітики даних
Мацевич Денис Володимирович

Рецензент: кандидат психологічних наук, доцент, доцент
кафедри інформаційних технологій та аналітики даних
Зубенко Ігор Ростиславович

РОБОТА ДОПУЩЕНА ДО ЗАХИСТУ
Завідувач кафедри інформаційних технологій та аналітики даних _____
проф., д.е.н. Кривицька О.Р.
Протокол №10 від 28 травня 2025 р.

Острог, 2025

АНОТАЦІЯ
кваліфікаційної роботи
на здобуття освітнього ступеня бакалавра

Тема: *Імплементація стандарту WCAG (Web Accessibility) для системи управління бібліотекою з онлайн каталогом та резервуванням книг*

Автор: *Фасоля Валентин Олегович*

Науковий керівник: *Мацевич Денис Володимирович*

Захищена «.....»..... 20__ року.

Пояснювальна записка до кваліфікаційної роботи: *81 (кількість сторінок роботи) с., 7 (кількість рисунків) рис., 0 (кількість таблиць) табл., 1 (кількість додатків) додатків, 18 (кількість джерел) джерел.*

Ключові слова: *ДОСТУПНІСТЬ, WCAG 2.1, БІБЛІОТЕЧНА СИСТЕМА, VUE, LIGHTHOUSE*

Короткий зміст праці:

Кваліфікаційна робота присвячена аналізу та впровадженню вимог веб-доступності відповідно до стандарту WCAG 2.1 у систему управління бібліотекою з онлайн-каталогом. У роботі розглянуто проблематику доступу користувачів з інвалідністю до інформаційних систем, а також визначено релевантні критерії за принципами POUR (сприйнятність, керованість, зрозумілість, надійність). На базі фреймворку Vue.js реалізовано покращення інтерфейсу для забезпечення доступу за допомогою клавіатури, візуальних фокусів, альтернативних текстів і семантичної структури. Проведено перевірку за допомогою інструменту Lighthouse. Результати тестування свідчать про значне покращення рівня доступності інтерфейсу відповідно до рівня AA стандарту WCAG.

Keywords: ACCESSIBILITY, WCAG 2.1, LIBRARY SYSTEM, VUE, LIGHTHOUSE

Summary of the work:

The bachelor's qualification thesis is dedicated to the analysis and implementation of web accessibility requirements in accordance with the WCAG 2.1 standard in a library management system with an online catalog. The work addresses the issue of access for users with disabilities to information systems and defines relevant criteria based on the POUR principles (Perceivable, Operable, Understandable, Robust). Using the Vue.js framework, the interface was improved to ensure access via keyboard navigation, visible focus indicators, alternative texts, and semantic structure. Testing was conducted using the Lighthouse tool. The results demonstrate a significant improvement in the accessibility level of the interface, reaching WCAG AA compliance.

(niɔnuc aɛmopa)

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. ЗАГАЛЬНІ ПОЛОЖЕННЯ	10
1.1. Визначення мети та завдань роботи	10
1.2. Короткий опис предметної області	12
1.3. Постановка задачі	20
Висновки до розділу 1	22
РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	24
2.1. Визначення веб-доступності та огляд стандарту WCAG 2.1	24
2.2. Принципи POUR (Perceivable, Operable, Understandable, Robust)	28
2.3. Рівні відповідності (A, AA, AAA)	32
2.4. Міжнародна та українська нормативно-правова база щодо доступності	37
2.5. Приклади порушень доступності на практиці	41
Висновки до розділу 2	49
РОЗДІЛ 3. ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	51
3.1. Архітектура системи та загальний підхід	51
3.2. Реалізація принципів WCAG у системі	56
3.3. Використання інструментів перевірки доступності (Lighthouse)	71
3.4. Юзабіліті-тестування із залученням користувачів	72
Висновки до розділу 3	75
ВИСНОВКИ	76
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ	78
ДОДАТКИ	81

ВСТУП

У сучасному цифровому суспільстві веб-ресурси відіграють ключову роль у забезпеченні доступу до інформації, послуг та сервісів. Однією з категорій таких ресурсів є бібліотечні системи з онлайн-каталогами та можливістю резервування книг. Вони значно спрощують взаємодію користувачів з бібліотеками, дозволяючи здійснювати пошук, перегляд наявності видань та бронювання книг у будь-який час.

Однак значна частина таких веб-систем залишається недоступною для користувачів з інвалідністю, що порушує їхні права на рівний доступ до інформації згідно з міжнародними та національними нормами. Забезпечення цифрової інклюзії — це не лише етичний обов'язок, а й вимога чинного законодавства багатьох країн, включаючи Україну.

Стандарт WCAG (Web Content Accessibility Guidelines) є загальноприйнятим набором рекомендацій, що визначають вимоги до доступності веб-контенту.[1] Імплементация WCAG забезпечує створення інтерфейсів, якими можуть користуватися люди з порушеннями зору, слуху, моторики, когнітивними порушеннями тощо.

У контексті державної цифровізації та діджиталізації освітнього середовища, питання доступності бібліотечних систем набуває особливої актуальності. Бібліотеки, як центри зберігання знань, мають бути відкритими для всіх категорій населення, незалежно від їхніх фізичних можливостей.

У цьому контексті дослідження, спрямоване на імплементацию стандарту WCAG у систему управління бібліотекою, є актуальним і суспільно значущим. Така робота дозволяє не лише підвищити якість програмного забезпечення, а й зробити внесок у формування інклюзивного цифрового середовища.

Мета дослідження – впровадження принципів і рекомендацій стандарту WCAG (Web Content Accessibility Guidelines) у веб-інтерфейс системи управління бібліотекою з онлайн-каталогом та резервуванням книг.

Для досягнення поставленої мети необхідно вирішити такі **завдання**:

1. Провести аналіз нормативно-правової бази та міжнародних стандартів у сфері веб-доступності, зосередившись на стандарті WCAG 2.1.
2. Визначити специфічні потреби користувачів бібліотечних систем, які мають обмеження зору, слуху чи моторики.
3. Оцінити поточний стан доступності обраної бібліотечної системи та визначити її недоліки згідно з критеріями WCAG.
4. Розробити та реалізувати зміни у фронтенд-частині системи (на прикладі фреймворку Vue.js), які відповідають вимогам рівня доступності AA.
5. Провести перевірку реалізованих рішень за допомогою автоматизованих інструментів тестування доступності.
6. Проаналізувати отримані результати, визначити ефективність імплементації.

Об'єкт дослідження – головна сторінка вебінтерфейсу системи електронної бібліотеки, розроблена на основі фреймворку Vue.js.

Предмет дослідження – методи та засоби створення доступного веб-інтерфейсу інформаційної системи з акцентом на проєктування, верстку та реалізацію головної сторінки відповідно до стандартів WCAG 2.1. У дослідженні розглядаються аспекти доступності, зручності користування, структурування вмісту, підтримки навігації за допомогою клавіатури, коректного маркування елементів інтерфейсу та візуальної відповідності.

Практична цінність розробленого інтерфейсу полягає у створенні доступного, сучасного та інтуїтивно зрозумілого вебінтерфейсу для користувачів електронної бібліотеки. Основні реалізовані можливості:

- адаптивний інтерфейс для різних типів пристроїв (десктопи, планшети, смартфони);
- логічно структурована головна сторінка з розміщенням ключових елементів: логотипу, навігації, блоку з інформацією про систему, списку книг, футера тощо;
- реалізована підтримка навігації через клавіатуру (Alt+1, Alt+2 тощо);
- стилізація фокусованих елементів для користувачів, які не користуються мишею;
- збереження читабельності та базових принципів веб-доступності відповідно до вимог WCAG 2.1.

Сторінка є частиною більшого проєкту, однак її якісна реалізація як доступного компонента є важливою складовою для забезпечення рівного доступу до бібліотечних ресурсів для всіх категорій користувачів.

РОЗДІЛ 1.

ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1. Визначення мети та завдань роботи

Основною метою даної роботи є розробка головної сторінки інтерфейсу веб-системи електронної бібліотеки, яка відповідатиме вимогам сучасної фронтенд-розробки, з урахуванням стандартів веб-доступності згідно з WCAG 2.1. Врахування принципів доступності у веб-інтерфейсі забезпечує рівний доступ до функціоналу системи для всіх користувачів, зокрема для осіб із порушенням зору, моторики чи когнітивних функцій. Тому головна сторінка повинна бути не лише функціональною, логічною та зручною, а й інклюзивною.

Розробка головної сторінки — це не лише візуальний дизайн і структура, а й забезпечення семантичної коректності, адаптивності, чіткої навігації та можливості взаємодії з системою без використання миші. Це особливо важливо для електронних бібліотек, що передбачають широку аудиторію користувачів із різним рівнем цифрових навичок і фізичних можливостей.

Згідно з вимогами сучасної фронтенд-розробки, створення доступного веб-інтерфейсу передбачає використання таких технологій, як HTML (для визначення семантики сторінки), CSS (для стилізації елементів інтерфейсу) та JavaScript/Vue.js (для реалізації логіки взаємодії користувача з інтерфейсом). Проте ключовим чинником є не лише вибір технологій, а й дотримання принципів доступності під час їх використання.

Зокрема, відповідність стандартам WCAG 2.1 досягається через чітке фокусування активних елементів, підтримку клавіатурної навігації, коректне використання HTML-розмітки, логічну структуру контенту, а також семантичні та aria-атрибути для допоміжних технологій (екранних читачів тощо). Застосування цих принципів дозволяє значно розширити потенційне коло користувачів і зробити систему більш інклюзивною.

Для реалізації поставленої мети необхідно вирішити низку практичних завдань, а саме:

- Проаналізувати стандарти веб-доступності (зокрема WCAG 2.1) та визначити ключові вимоги до побудови доступного інтерфейсу. Особливу увагу було приділено чотирьом базовим принципам: сприйнятність, керованість, зрозумілість і надійність.
- Опрацювати інструменти, фреймворки та бібліотеки, необхідні для розробки вебінтерфейсу, зокрема Vue CLI, Vue.js 3, HTML5, CSS3 та інші супутні засоби, які забезпечують сучасну компонентну розробку.
- Реалізувати підтримку клавіатурної навігації за допомогою гарячих клавіш (Alt+1, Alt+2 тощо), що дозволяє швидко переходити між основними областями сторінки (головний вміст, навігація, футер), що критично важливо для користувачів, які не користуються мишею.
- Створити візуальні стилі для фокусованих елементів (наприклад, кнопок і посилань), які виділяються під час навігації клавішами Tab та Shift+Tab, що є вимогою WCAG 2.1 (критерій 2.4.7 — «Видимий фокус»).
- Забезпечити логічну ієрархію заголовків та розділів сторінки через правильне використання тегів `<h1>`–`<h6>`, `<main>`, `<nav>`, `<footer>`, `<section>`, що дозволяє зручно орієнтуватися вмістом через допоміжні технології.
- Виконати тестування доступності сторінки за допомогою таких інструментів, як Lighthouse, що дозволяють перевірити відповідність коду стандартам доступності та виявити критичні помилки (наприклад, відсутність альтернативного тексту, порушення кольорової контрастності, некоректне фокусування тощо).
- Оцінити інклюзивність, зручність та адаптивність розробленого інтерфейсу — тобто наскільки легко ним користуватися різним групам користувачів, включно з тими, хто працює на екранах різної ширини або за допомогою скрінрідерів.

Таким чином, виконання вищезазначених завдань дозволяє створити головну сторінку бібліотечної системи, яка не лише візуально приваблива та функціональна, а й доступна для широкого кола користувачів, включаючи людей з інвалідністю. Цей підхід відповідає сучасним вимогам інклюзивного дизайну, що має особливу вагу в контексті створення публічних освітніх систем, зокрема електронних бібліотек.

1.2. Короткий опис предметної області

У сучасному суспільстві доступ до інформації та цифрових сервісів стає фундаментальним правом людини. Це особливо актуально в контексті електронних бібліотек і онлайн-каталогів, які відіграють ключову роль у забезпеченні рівного доступу до знань, культури та освіти. Веб-системи бібліотек, зокрема ті, що надають можливість перегляду, фільтрації, бронювання та збереження літератури, повинні бути однаково зручними для всіх користувачів, незалежно від їхніх фізичних або когнітивних можливостей.

Однак, практика показує, що веб-застосунки часто не враховують специфічні потреби людей з інвалідністю. Розуміння цих потреб — перший крок до створення по-справжньому інклюзивного цифрового середовища. У цьому підрозділі буде детально проаналізовано, які саме труднощі можуть виникати в користувачів з різними формами інвалідності при роботі з бібліотечними веб-системами, а також які функціональні, технічні та дизайнерські рішення можуть задовольнити ці потреби.

Категорії користувачів з інвалідністю та їхні потреби.

1. Люди з порушенням зору

Це одна з найбільш чисельних категорій користувачів з інвалідністю, яка охоплює:

- повну сліпоту;
- часткову втрату зору;

- кольорову сліпоту (дальтонізм);
- нечітке або тунельне бачення.

Основні потреби:

- Озвучення вмісту сторінки за допомогою скрінрідерів;
- Високий контраст між текстом і фоном, щоб полегшити читання для людей із залишковим зором;
- Масштабування тексту без втрати структури сторінки;
- Альтернативні тексти для зображень (атрибут alt);
- Семантична розмітка, яка дозволяє навігацію не по візуальному вигляду, а по логічній структурі сторінки.

Проблеми в бібліотечних системах:

- Відсутність опису до обкладинок книг;
- Списки літератури оформлені <div>-елементами замість /;
- Фільтри подані через кнопки без aria-label;
- Поганий контраст кольорів у дизайні (наприклад, темно-сірий текст на світло-сірому фоні).

2. Люди з порушеннями слуху

Ці користувачі можуть мати:

- часткову або повну глухоту;
- труднощі зі сприйняттям мовлення;
- потребу у візуальній інформації.

Основні потреби:

- Текстові альтернативи до мультимедійного контенту, якщо сайт використовує відео чи аудіофайли;
- Інформація у зрозумілому, логічному вигляді, з уникненням неоднозначності;

- Візуальні сповіщення про події в системі (наприклад, підтвердження, що книга зарезервована).

3. Люди з порушеннями моторики

Ці користувачі можуть мати:

- часткову або повну втрату контролю над руками;
- труднощі з координацією;
- тремор, спастичність, слабкість.

Основні потреби:

- Повна клавіатурна навігація — можливість керувати всіма функціями без миші;
- Мінімізація потреби у точності: уникнення дрібних чекбоксів, випадаючих меню без затримки;
- Можливість використання адаптивних пристроїв: джойстики, перемикачі, екранні клавіатури.

4. Люди з когнітивними порушеннями

Це група, що охоплює користувачів з:

- труднощами концентрації;
- розладами уваги, пам'яті;
- дислексією, аутизмом, емоційними порушеннями.

Основні потреби:

- Простий, лаконічний, послідовний інтерфейс;
- Передбачувана структура сторінки;
- Зрозумілі тексти без термінології чи складних конструкцій;
- Мінімізація навантаження на короткотривалу пам'ять;

- Підказки, підкріплення діалогів повідомленнями (наприклад, при помилці у формі — чітко зазначити, що саме введено неправильно).

Проблеми в бібліотечних системах:

- Велику кількість елементів на одній сторінці;
- Недостатньо підписані поля вводу;
- Складні слова в описі книг або фільтрах (“метааналіз”, “гуманітаристика” тощо);
- Раптові переходи між сторінками без підтвердження.

5. Люди з комбінацією порушень

Не слід забувати, що деякі користувачі можуть одночасно мати кілька типів обмежень. Наприклад, людина похилого віку може мати:

- поганий зір;
- труднощі з моторикою;
- знижену здатність до адаптації в цифровому середовищі.

Тому системи мають підтримувати адаптивність до широкого спектра потреб і враховувати принципи універсального дизайну, що передбачають зручність використання для всіх, без потреби у модифікації або спеціалізації.

Типові інтерфейси бібліотечних систем

Розглянемо, з яких функціональних та візуальних компонентів зазвичай складається веб-інтерфейс бібліотечної системи з онлайн-каталогом та функцією бронювання книг. Кожен із цих інтерфейсів має свою специфіку з точки зору реалізації доступності, і для кожного можна виділити типові проблеми й рекомендації згідно з стандартом WCAG 2.1.

Загальна структура інтерфейсу бібліотечної системи

Типова веб-система для управління бібліотекою включає такі основні функціональні зони:

1. Головна сторінка
2. Пошук і фільтрація літератури
3. Сторінка результатів пошуку / перегляду каталогу
4. Картка книги
5. Форма авторизації / реєстрації
6. Особистий кабінет користувача
7. Кошик бронювання
8. Адміністративна панель (для бібліотекарів)
9. Сторінки з контактною інформацією, правилами користування

1. Головна сторінка

Значення для користувача:

Це стартова точка взаємодії з системою. Вона часто містить короткий опис функціоналу, панель навігації, поле для пошуку, банери, новини бібліотеки або рекомендації.

Типові елементи:

- Меню навігації
- Кнопки “Пошук”, “Увійти”, “Зареєструватись”
- Промо-блоки (наприклад: “Нова література”, “Рекомендоване”)
- Новини або події

Поширені проблеми з доступністю:

- Відсутність чіткої фокусної рамки при переході клавіатурою
- Незрозумілі текстові альтернативи до іконок
- Контрастність між фоном та текстом (особливо в банерах)

WCAG-рішення:

- Використання чіткої семантики (<nav>, <main>, <header>)
- Фокус-індикатори для всіх інтерактивних елементів
- Тестування з скрінрідером: чи правильно читається порядок елементів

2. Інтерфейс пошуку та фільтрації

Значення:

Дозволяє знайти потрібну літературу серед великого обсягу даних. Часто має розширений фільтр: за автором, жанром, роком, наявністю, типом носія (електронна/друкована книга).

Типові елементи:

- Поле пошуку з кнопкою
- Випадаючі списки, чекбокси
- Поля з автозаповненням

Проблеми:

- Відсутність лейблів для форм
- Чекбокси та радіокнопки не мають описів
- Неможливо користуватись фільтрами клавіатурою
- Недоступні повідомлення про помилки у запитах

WCAG-рішення:

- Пов'язування <label> з відповідним полем через атрибут for
- Уникання placeholder як єдиного способу інформування користувача
- Створення підказок і текстів-інструкцій до полів
- Реалізація ARIA-полей: aria-live, aria-describedby

3. Каталог / Сторінка результатів пошуку

Значення:

Це головна зона взаємодії користувача зі змістом бібліотеки. Тут відображається список знайдених книг, короткі картки, кнопки перегляду або бронювання.

Типові елементи:

- Список карток книг
- Кнопки “Переглянути”, “Забронювати”
- Пейджинг (навігація сторінками)
- Фільтрація результатів (іноді збоку)

Проблеми:

- Неправильна семантична розмітка (наприклад, картки в `<div>` без `role="list"` або `role="article"`)
- Відсутність тексту на кнопках (використання лише іконок)
- Кнопки не мають опису, що саме буде зроблено (наприклад, “Забронювати книгу: Назва”)

WCAG-рішення:

- Використання атрибутів ARIA: `aria-label`, `aria-labelledby`
- Рольові теги (`<article>`, `<section>`)
- Можливість переходу клавішами TAB між усіма результатами

4. Картка книги

Значення:

Детальна інформація про обраний ресурс: автор, рік, жанр, анотація, статус, доступність, кнопка “Забронювати”.

Типові проблеми:

- Відсутність текстового дублювання зображення обкладинки
- Неможливо дізнатись, чи доступна книга — інформація подається лише кольором (наприклад, зелений — доступна, червоний — ні)
- Відсутність логічного порядку фокусування

WCAG-рішення:

- Атрибути alt для всіх зображень
- Текстовий статус (“доступна”, “взята”)
- Кнопка з чітким описом: `aria-label="Забронювати книгу: Назва книги"`

5. Форма авторизації / реєстрації

Типові проблеми:

- Поля не підписані
- Немає повідомлень про помилки або вони не озвучуються скрінрідером
- Кнопки “Увійти” чи “Реєстрація” недоступні через клавіатуру

WCAG-рішення:

- Валідація в реальному часі з повідомленням через `aria-live`
- Стандартні елементи форм (`<input>`, `<label>`, `<fieldset>`)
- Доступність кнопок через `role="button"` та `tabindex="0"`

6. Особистий кабінет

Зміст:

Інформація про обліковий запис, список заброньованих книг, історія замовлень, повідомлення, можливість скасувати бронювання.

Типові недоліки:

- Вкладки реалізовані лише візуально, без ARIA-ролей
- Поганий фокус-контроль

WCAG-рішення:

- Вкладки з `role="tablist"`, `role="tab"`, `aria-selected`
- Всі повідомлення про події — через `aria-live`

7. Кошик бронювання

Проблеми:

- Елементи не підписані
- Неможливо редагувати/видалити замовлення за допомогою клавіатури

Рішення:

- Чітке озвучення дій скрінрідером
- Кнопки типу “Видалити книгу” з `aria-label`

1.3. Постановка задачі

Оцінка вихідного стану проекту з точки зору доступності

На цьому етапі кваліфікаційної роботи важливо провести детальний аналіз вихідного стану бібліотечної веб-системи, реалізованої під час попередніх етапів навчання (курсової роботи та практики), з точки зору її відповідності стандартам веб-доступності WCAG 2.1. Метою є виявлення недоліків, бар'єрів і типових помилок, які можуть створювати труднощі для користувачів з інвалідністю при взаємодії з інтерфейсом, а також підготувати ґрунт для подальшої імплементації рекомендацій стандарту.

Загальний огляд реалізованої системи

Бібліотечна система створена з використанням фреймворку Vue.js, що дозволяє реалізовувати компонентно-орієнтовану архітектуру. У межах попередньої курсової роботи був розроблений дизайн інтерфейсу з використанням Figma, а під час практики — реалізовано основну візуальну частину проєкту, включаючи такі ключові сторінки та компоненти:

- Головна сторінка — з навігаційним меню, заголовком, інформаційним блоком та цитатою дня;
- Сторінка книги — деталізована інформація про окремий об'єкт у каталозі;
- Форма авторизації — дозволяє користувачам входити до системи;
- Сайдбар (бічна панель) — елемент навігації або фільтрації;
- Футер і хедер — основні структурні компоненти сторінок;
- Букліст — перелік книг, можливо з кнопками взаємодії або перегляду деталей.

Поточний стан доступності

На момент розробки веб-системи жодні спеціалізовані практики з доступності не були реалізовані. Це означає, що:

- відсутні семантичні теги HTML5 (`<main>`, `<nav>`, `<header>`, `<footer>`, `<section>`);
- не враховано логіку навігації через клавіатуру (не всі елементи, ймовірно, доступні через TAB);
- відсутні текстові альтернативи для зображень (наприклад, обкладинок книг);
- інтерфейс не підтримує екранні зчитувачі (немає `aria-*` атрибутів);
- кольорова схема, шрифти, і контрастність не були перевірені на відповідність рекомендаціям WCAG;

Приклади очікуваних проблем

Хоча формально жодні автоматизовані інструменти оцінки ще не були застосовані, можна передбачити наявність ряду критичних бар'єрів:

- Неможливість повноцінної навігації без миші — для людей із моторними порушеннями.
- Незрозумілі елементи інтерфейсу — якщо кнопки мають лише іконки без підпису.
- Інформація, яка подається лише через колір — наприклад, статус книги (доступна / недоступна).
- Відсутність структури заголовків (H1-H6) — що критично для скрінрідерів.
- Форми без валідації й повідомлень — користувач з інвалідністю не дізнається, що ввів неправильно.

Отже, аналізуючи стан бібліотечної системи до впровадження стандартів доступності, можна зробити висновок, що:

- інтерфейс не враховує потреби людей з інвалідністю;
- порушено базові принципи WCAG 2.1;
- система є складною у користуванні для людей з обмеженими можливостями;
- необхідно переглянути структуру сторінок, семантику, а також впровадити ARIA-атрибутів, підтримку навігації з клавіатури, контрастності, масштабованості шрифтів тощо.

Висновки до розділу 1

У першому розділі було визначено основну мету роботи — впровадження принципів веб-доступності відповідно до стандарту WCAG 2.1 в інтерфейс системи управління бібліотекою. Для досягнення цієї мети було сформульовано такі завдання: визначити актуальні вимоги WCAG 2.1, встановити рівень поточної відповідності інтерфейсу цим вимогам, а також спланувати конкретні технічні кроки з адаптації інтерфейсу до потреб користувачів з порушеннями зору, моторики чи сприйняття.

У процесі аналізу було обґрунтовано доцільність впровадження таких елементів, як логічна структура розмітки, підтримка клавіатурної навігації, видимий фокус, альтернативний текст, коректне маркування форм тощо. Отримані результати стали основою для подальшої реалізації змін у проєкті відповідно до обраного рівня доступності.

РОЗДІЛ 2.

ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1. Визначення веб-доступності та огляд стандарту WCAG 2.1

Веб-доступність — це поняття, яке охоплює цілу низку практик, орієнтованих на створення інклюзивного досвіду користування веб-ресурсами для всіх людей, незалежно від їх фізичних, когнітивних чи сенсорних можливостей.[2] Веб-доступність передбачає, що кожен користувач, у тому числі й особи з інвалідністю, має змогу ефективно взаємодіяти з веб-контентом, отримувати доступ до інформації та виконувати необхідні дії на веб-сайті або в веб-додатку. Веб-доступність охоплює не лише звичайні аспекти навігації та перегляду інформації, а й забезпечує можливість взаємодії з цифровими продуктами незалежно від того, чи має людина обмеження за зрением, слухом, руховими функціями або когнітивними здібностями.

Ключові компоненти веб-доступності

1. Технічні аспекти доступності. Один з основних елементів веб-доступності полягає в тому, щоб контент веб-ресурсу був сприйнятний і доступний для всіх користувачів, включаючи тих, хто використовує допоміжні технології, такі як екранні читалки тощо. Веб-ресурси повинні бути спроектовані та реалізовані так, щоб їх було можливо налаштувати для індивідуальних потреб користувачів.
2. Структура і організація контенту. Веб-доступність передбачає, що контент на сайті має бути структурований таким чином, щоб його було легко зрозуміти й сприймати. Структурні елементи, такі як заголовки, абзаци, списки та посилання, мають бути правильно розмічені та організовані. Веб-сайт або веб-додаток повинен забезпечувати чітку навігацію з логічною та передбачуваною організацією елементів інтерфейсу, що дозволяє користувачам орієнтуватися без зайвих зусиль. Для людей з когнітивними

порушеннями важливо, щоб інформація на сайті була подана максимально чітко й зрозуміло, без зайвих складностей.

3. Управління та взаємодія з контентом. Веб-доступність передбачає також, що користувачі повинні мати можливість взаємодіяти з веб-ресурсами, як за допомогою стандартної миші, так і за допомогою клавіатури або інших спеціалізованих пристроїв. Наприклад, люди з обмеженими можливостями руху можуть використовувати клавіатуру для навігації по сайту, тому важливо, щоб елементи керування були доступні для таких способів взаємодії. Кожен елемент на сторінці має бути зручний для доступу та використання незалежно від того, який пристрій або технологію використовує користувач.
4. Альтернативні засоби для людей з обмеженнями органів чуття. Велике значення для забезпечення веб-доступності має наявність альтернативних форм контенту, які можуть бути корисними для людей з обмеженнями органів чуття. Для осіб із порушеннями зору важливо наявність текстових описів зображень та інших мультимедійних елементів, що дозволяють передати інформацію в іншому форматі. Для людей з порушеннями слуху важливим є наявність субтитрів до відео та відео з жестовою мовою. Це дозволяє забезпечити рівний доступ до інформації та контенту для осіб з різними видами інвалідності.

Важливість веб-доступності

Забезпечення веб-доступності є важливим не тільки для осіб з інвалідністю, але й для широкого кола користувачів. Інтерфейси, що відповідають вимогам доступності, зазвичай є більш зрозумілими, простими у використанні та зручними для всіх. Наприклад, чітка організація контенту, можливість навігації за допомогою клавіатури, оптимізація шрифтів і кольорів роблять веб-ресурси доступнішими для людей похилого віку або тих, хто використовує мобільні пристрої з маленьким екраном.

Крім того, з огляду на постійний розвиток цифрових технологій та глобалізацію інтернету, забезпечення доступності є важливим аспектом у створенні справедливих та рівних умов для всіх користувачів. Веб-ресурси, які не відповідають вимогам доступності, обмежують доступ до інформації та послуг, що може призвести до соціальної ізоляції людей з інвалідністю.

Проблеми та виклики веб-доступності

Незважаючи на численні зусилля з покращення веб-доступності, існує багато проблем, які потребують вирішення. Однією з найбільших проблем є відсутність належної обізнаності серед розробників і дизайнерів про важливість доступності. Багато веб-ресурсів продовжують містити елементи, які не відповідають вимогам WCAG, наприклад, неправильні теги для зображень, недостатній контраст тексту і фону, або відсутність можливості навігації за допомогою клавіатури.

Ще однією проблемою є недостатнє тестування веб-ресурсів з точки зору доступності, оскільки багато компаній не проводять оцінку доступності на всіх етапах розробки. Це часто призводить до того, що веб-сайти не відповідають вимогам стандарту, і користувачі з інвалідністю не можуть повноцінно користуватися такими ресурсами.

Основи стандарту WCAG 2.1

Web Content Accessibility Guidelines (WCAG) — це міжнародні рекомендації, розроблені W3C (World Wide Web Consortium) для забезпечення доступності веб-контенту для всіх користувачів, зокрема осіб з інвалідністю.[3] Стандарт WCAG розроблений з метою встановлення вимог до веб-ресурсів, що дозволяють зробити їх доступними для максимально широкого кола користувачів, включаючи людей з різними типами обмежень: зору, слуху, рухових функцій і когнітивними порушеннями.

Остання версія стандарту — **WCAG 2.1** була опублікована в червні 2018 року. Вона є оновленням попередньої версії WCAG 2.0, яка була випущена в 2008 році.

Основна мета WCAG 2.1 полягає в тому, щоб зробити інтернет доступним для ще більшої кількості користувачів, зокрема осіб з порушеннями зору, слуху, руху та когнітивними обмеженнями. Він містить чотири основні принципи, які лежать в основі забезпечення доступності веб-контенту.

WCAG 2.1 доповнює WCAG 2.0 новими вимогами, орієнтуючись на потреби користувачів з різними типами інвалідності, зокрема осіб з порушеннями рухової активності, когнітивними та навчальними обмеженнями. Ось основні нововведення:

1. Вимоги для осіб з порушеннями рухової активності

Веб-сайти та додатки повинні бути оптимізовані для людей, які не можуть використовувати мишу, але можуть взаємодіяти з інтерфейсом за допомогою клавіатури або інших альтернативних засобів введення. Включення функцій, таких як можливість переміщувати фокус за допомогою клавіатури, є важливими для забезпечення доступності.

2. Покращення доступності для користувачів з когнітивними та навчальними обмеженнями

У WCAG 2.1 додано вимоги для покращення доступності контенту для людей з когнітивними порушеннями. Це включає надання зрозумілих пояснень для складних концепцій, використання простих мовних формулювань і запобігання інформаційним навантаженням. Принципи, які допомагають розробникам створювати зрозумілі та інтуїтивно зрозумілі інтерфейси, стають основою для цієї категорії користувачів.

3. Покращення доступності для мобільних пристроїв

Оскільки мобільні пристрої стали основними для багатьох користувачів, WCAG 2.1 також звертає увагу на адаптацію інтерфейсів для маленьких екранів. Цей аспект включає забезпечення правильного розміщення елементів інтерфейсу, їх розмірів та відстаней для мобільних пристроїв, щоб користувачі могли зручно використовувати веб-ресурси на смартфонах і планшетах.

Ключові характеристики WCAG 2.1

1. Логічність та послідовність

Один із важливих аспектів доступності — це логічна та послідовна структура веб-ресурсу. Користувачі повинні мати змогу без проблем орієнтуватися в контенті та переходити від одного розділу до іншого. Це досягається за допомогою чіткої організації та структурування контенту: правильного використання заголовків, абзаців, списків і зв'язків.

2. Мультиформатність

Веб-контент має бути доступним не лише в текстовому форматі, а й у мультимедійних форматах, які враховують потреби користувачів з різними обмеженнями. Це означає, що для відео повинні бути доступні субтитри, для зображень — альтернативні текстові описи, а для складних інтерфейсів — можливість управління за допомогою альтернативних засобів.

3. Тестування та перевірка доступності

Оцінка доступності веб-ресурсів є важливим етапом у процесі розробки. Для цього розробники використовують різноманітні інструменти для тестування відповідності стандарту WCAG, щоб виявити й усунути можливі проблеми з доступністю на ранніх етапах створення сайту або додатку.

Стандарт WCAG 2.1 є потужним інструментом для розробників, дизайнерів і організацій, що прагнуть забезпечити доступність своїх веб-ресурсів. Врахування вимог WCAG 2.1 дозволяє не лише полегшити доступ до інтернет-ресурсів для людей з обмеженнями, а й створює більш інклюзивний досвід для всіх користувачів. Оскільки інтернет стає все важливішим інструментом у різних аспектах життя, від освіти до роботи та розваг, забезпечення доступності веб-ресурсів стає необхідністю для кожної організації, яка прагне бути соціально відповідальною та інклюзивною.

2.2. Принципи POUR (Perceivable, Operable, Understandable, Robust)

Принципи **POUR** (Perceivable, Operable, Understandable, Robust) є основою стандарту **WCAG 2.1** і являють собою чотири базові вимоги до веб-доступності, які забезпечують рівний доступ до інформації та функцій для всіх користувачів. Ці

принципи виступають як методологічні орієнтири для розробників, дизайнерів і організацій, що прагнуть створити інклюзивні, доступні веб-ресурси. Усі ці принципи мають велике значення для забезпечення доступності веб-контенту, адже вони націлені на різні аспекти взаємодії з цифровим середовищем. Розглянемо кожен принцип окремо.

1. Perceivable (Спостережуваність)

Принцип **Perceivable** означає, що користувачі повинні мати можливість сприймати всю інформацію на веб-ресурсі, незалежно від своїх обмежень. Це включає забезпечення того, щоб контент був доступний через різні канали сприйняття — зір, слух, тактильне сприйняття та інші способи сприймання, які можуть бути доступні користувачам з інвалідністю. Цей принцип фокусується на наданні достатньої кількості інформації у різних форматах, щоб кожен користувач мав можливість отримати доступ до важливої інформації, що надається на веб-сайті.

1. Текстові альтернативи для мультимедійного контенту

Наприклад, відео або зображення повинні мати **альтернативні текстові описи** (alt-теги). Це дозволяє людям з обмеженнями зору використовувати екранні читалки для отримання інформації, що надається у вигляді зображень чи відео. У випадку відео також повинні бути доступні **субтитри**, а для складних відео-контентів — **звукові описи**. Субтитри не тільки полегшують сприйняття інформації для людей з порушеннями слуху, але й роблять її доступною для людей, які знаходяться в умовах, де вони не можуть включити звук (наприклад, у громадських місцях).

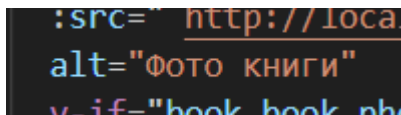


Рис. 2.1. Приклад використання alt-тегу

Джерело: розроблено автором

2. Контраст між текстом та фоном

Зображення чи текст на веб-сторінці повинні мати **достатній контраст** між текстом і фоном, що допомагає людям з обмеженим зором або певними типами кольорової сліпоти краще сприймати інформацію. Наприклад, текст темного кольору на світлому фоні зазвичай є набагато легшим для сприйняття, ніж бліді або півтоновані варіанти.

Example Example Example

Рис. 2.2. Приклад правильного підбору кольору з достатнім контрастом,
співвідношення 12.6:1

Джерело: розроблено автором

3. Гнучкість у поданні інформації

Веб-контент має бути адаптований під різні розміри екрану та різні пристрої. Це включає забезпечення того, щоб текст та інші елементи на сторінці автоматично змінювались, коли розмір екрану змінюється, наприклад, на мобільних пристроях. Користувачі з вадами зору, які використовують масштабування тексту або інші методи, мають мати можливість коректно переглядати веб-ресурс, не втрачаючи важливу інформацію.

2. Operable (Операбельність)

Принцип **Operable** передбачає, що всі елементи на веб-сайті повинні бути доступні для взаємодії та використання. Це включає можливість здійснювати всі необхідні дії на сайті без зайвих труднощів, що особливо важливо для людей з фізичними обмеженнями, такими як порушення моторики або рухливості. Наприклад, важливо, щоб веб-сайт підтримував взаємодію не лише за допомогою миші, але й за допомогою клавіатури, голосових команд або спеціалізованих пристроїв.

1. Навігація за допомогою клавіатури

Усі елементи на веб-сторінці повинні бути доступні для навігації за допомогою клавіатури. Це особливо важливо для людей, які не можуть

використовувати мишу, а також для тих, хто використовує клавіатурні альтернативи, такі як спеціальні пристрої для управління. Це включає кнопки, посилання, меню, форми тощо.

2. Уникнення використання швидких рухів

Веб-сайти повинні уникати використання елементів, які можуть спричинити проблеми у користувачів, особливо у тих, хто має епілепсію або інші неврологічні порушення. Наприклад, анімації, що швидко змінюються, можуть бути небезпечними, тому важливо дати можливість вимкнути їх для користувачів.

3. Understandable (Зрозумілість)

Принцип **Understandable** означає, що контент веб-сайту має бути легким для розуміння. Це включає не лише подачу інформації зрозумілим та чітким способом, але й забезпечення того, щоб користувачі могли інтерпретувати елементи інтерфейсу без особливих труднощів. Простота і ясність є ключовими для людей з когнітивними порушеннями.

1. Простота мовлення

Мова веб-контенту повинна бути простою та прямою. Важливо уникати складних термінів, складних речень, а також жаргону, якщо це не є необхідним. Простий і зрозумілий стиль дозволяє користувачам швидко орієнтуватися в інформації.

2. Передбачуваність інтерфейсу

Веб-сторінка повинна бути передбачуваною для користувачів. Це означає, що елементи інтерфейсу мають бути логічно розташовані і працювати так, як користувачі очікують. Наприклад, кнопки мають виконувати відповідні дії, а не бути причиною плутанини чи непорозуміння.

3. Попередження про помилки

Важливо, щоб користувачі отримували чіткі повідомлення про помилки.

Наприклад, при заповненні форм потрібно вказувати, що саме не так з введеними даними, і як виправити помилку.

4. Robust (Надійність)

Принцип **Robust** означає, що веб-контент має бути стійким до змін та технологічних нововведень. Це передбачає, що веб-ресурс повинен підтримувати різні браузері, пристрої та допоміжні технології, забезпечуючи доступність на довгі роки. Надійність також включає можливість адаптації до нових технологій, які можуть з'явитися в майбутньому.

1. Сумісність з різними браузерами

Веб-контент має бути однаково доступним на різних браузерах і платформах. Це включає підтримку не лише популярних браузерів, таких як Chrome, Firefox, Safari чи Edge, але й менш відомих браузерів, які можуть використовуватися в певних регіонах або на специфічних пристроях.

2. Сумісність з допоміжними технологіями

Веб-ресурси повинні бути сумісні з такими допоміжними технологіями, як екранні читалки та подібні інструменти, що використовуються людьми з обмеженнями.

3. Правильний і стандартизований код

Використання сучасних стандартів веб-розробки (HTML, CSS, JavaScript) забезпечує, що контент буде доступний і для нових версій браузерів, і для застарілих версій, які все ще використовуються певною частиною користувачів.

2.3. Рівні відповідності (A, AA, AAA)

Стандарт WCAG 2.1 поділяє вимоги до доступності на три рівні відповідності: **A**, **AA** та **AAA**. [1] Кожен з цих рівнів має власні вимоги, що визначають, які саме аспекти веб-сайту повинні бути доступними для користувачів з різними типами обмежень. Рівні дозволяють веб-розробникам і дизайнерам визначити пріоритети і

поступово покращувати доступність ресурсів. Це також дозволяє організаціям оцінити рівень доступності їхніх веб-сайтів і вжити заходів для покращення доступу для всіх користувачів.

Рівень А (Основні вимоги)

Рівень А є базовим рівнем вимог до доступності і передбачає ті вимоги, виконання яких є необхідним для того, щоб сайт був хоча б частково доступним для осіб з обмеженими можливостями.[4] Якщо одна з вимог рівня А не виконується, весь веб-сайт або окремі його частини не можуть вважатися доступними для широкого кола користувачів, які потребують допоміжних технологій.

1. Визначення рівня А

Веб-сайт відповідає вимогам рівня А, якщо на ньому не присутні основні бар'єри, які можуть унеможливити використання сайту. Вимоги до рівня А є критичними, і їх виконання є першим кроком до створення доступного веб-контенту. Це включає необхідність забезпечення доступності для користувачів, які мають порушення зору, слуху, рухливості, когнітивні розлади або інші обмеження.

2. Ключові вимоги рівня А

На рівні А передбачається виконання найважливіших вимог, без яких користувачі не зможуть взаємодіяти з веб-контентом.[5] Наприклад:

- Альтернативний текст для всіх зображень і мультимедійних елементів, який дозволяє користувачам з порушеннями зору сприймати зміст графічного контенту через екранні читачі.
- Доступність навігації за допомогою клавіатури для тих, хто не може використовувати мишу або інші вказівні пристрої.
- Коректне позначення форм: для кожного поля форми має бути чітко вказано, яке саме введення очікується від користувача, щоб він міг правильно заповнити форму за допомогою доступних технологій.

3. Приклад порушення рівня А

Типовим порушенням вимог рівня А є відсутність альтернативного тексту для зображень або мультимедійного контенту. Це створює серйозні проблеми для користувачів, які мають порушення зору, оскільки вони не можуть сприймати зміст зображень через екранний читач. Якщо веб-сайт містить зображення, важливі для розуміння контексту (наприклад, графіки, діаграми, картинки продуктів чи послуг), то відсутність альтернативного тексту робить ці елементи неприступними для таких користувачів.

Якщо хоча б одна з вимог рівня А не виконується, веб-сайт вважається непридатним для користувачів з певними формами обмежень, а його загальний рівень доступності істотно знижується.

Рівень АА (Рекомендовані вимоги)

Рівень АА є більш високим, але одночасно й більш досяжним рівнем доступності, що передбачає вимоги, які мають покращити загальний доступ до веб-сайтів і зробити їх більш зручними для широкого кола користувачів, включаючи людей з помірними обмеженнями.

1. Визначення рівня АА

Рівень АА є рекомендованим стандартом, який слід виконувати для створення більш доступного контенту. Це включає покращення доступності в таких сферах, як контрастність тексту, використання шрифтів та надання інструкцій для користувачів із когнітивними порушеннями.[6] Більшість веб-сайтів мають відповідати вимогам рівня АА, оскільки саме цей рівень враховує основні потреби користувачів у повсякденному використанні та забезпечує баланс між доступністю й практичністю впровадження.

2. Ключові вимоги рівня АА

Основні вимоги рівня АА включають[7]:

- Достатній контраст між текстом та фоном. Це особливо важливо для людей з порушеннями зору, зокрема для тих, хто страждає на кольорову сліпоту або має ослаблений зір.
- Шрифти та інтерфейс. Тексти повинні бути читабельними й адаптуватися до змін розміру екрану (responsive fonts). Це дозволяє людям з порушеннями зору налаштовувати шрифт під свої потреби.
- Доступність мультимедійного контенту. Наприклад, наявність субтитрів для відео, що дозволяє користувачам з порушеннями слуху сприймати зміст контенту.
- Правильна структура веб-сторінки. Веб-сайт повинен мати чітку та передбачувану структуру, де важливі елементи мають бути позначені (наприклад, кнопки, форми, таблиці).

3. Приклад порушення рівня AA

Одним із порушень рівня AA може бути відсутність достатнього контрасту між текстом і фоном. Наприклад, якщо на веб-сторінці використовуються кольори, що погано виділяються на фоні (світло-сірий текст на світло-бежевому фоні), то людям із порушеннями зору або кольоровою сліпотою буде дуже складно читати текст.

4. Інші вимоги рівня AA

Важливою вимогою є адаптивність шрифтів, тобто можливість змінювати розмір тексту без втрати функціональності на сторінці. Це дозволяє користувачам з обмеженнями зору збільшувати шрифт для зручності читання.

Рівень AAA (Оптимальні вимоги)

Рівень AAA є найвищим рівнем відповідності стандарту WCAG 2.1 і включає найскладніші та найбільш детальні вимоги до доступності. Цей рівень орієнтований на забезпечення доступності для широкого кола користувачів, включаючи людей з серйозними порушеннями або дуже специфічними потребами. Проте не всі веб-сайти повинні відповідати рівню AAA, оскільки це може бути технічно складно

реалізувати на всіх сторінках. Проте для забезпечення максимальної доступності цей рівень є бажаним.[8]

1. Визначення рівня AAA

Рівень AAA визначає вимоги, які дозволяють забезпечити веб-доступність для користувачів із дуже специфічними обмеженнями, включаючи людей з серйозними когнітивними порушеннями, порушеннями слуху або зору, а також тих, хто має особливі потреби у використанні веб-контенту. Зазвичай вимоги рівня AAA стосуються спеціальних можливостей, таких як забезпечення максимального контрасту кольорів, підтримка різноманітних мовних формулювань, додаткові варіанти допомоги для користувачів з інтелектуальними порушеннями тощо.

2. Ключові вимоги рівня AAA

До вимог рівня AAA належать[9]:

- Персоналізація контенту. Веб-сайти повинні надавати можливість користувачам налаштовувати контент відповідно до їхніх індивідуальних потреб, наприклад, змінювати шрифт, кольорову палітру, або навіть вибрати текстовий формат замість графічного.
- Покращена підтримка для людей з когнітивними порушеннями. Наприклад, повинні бути доступні спрощені версії контенту або варіанти інтерфейсу, що спрощують навігацію і взаємодію з сайтом.
- Варіанти для мультимедійного контенту. Всі відео, аудіо та інші мультимедійні елементи повинні мати доступні субтитри, додаткові пояснення або переклад жестовою мовою, що дасть змогу користувачам з порушеннями слуху чи зору отримувати повний доступ до контенту.

3. Приклад порушення рівня AAA

Прикладом порушення рівня AAA може бути відсутність альтернатив для складних візуальних або аудіовізуальних елементів. Якщо на веб-сайті присутні відео, анімації або інші візуальні ефекти без додаткових субтитрів або пояснень для людей з порушеннями слуху чи зору, це вважається

порушенням стандарту WCAG на рівні AAA. Такі елементи можуть бути недоступними для користувачів, що потребують додаткових функцій для сприйняття інформації.

4. Інші важливі аспекти рівня AAA

Іншим прикладом вимог рівня AAA є можливість перекладу контенту на інші мови або на специфічні варіанти мови, наприклад, жестову мову, що дозволяє забезпечити максимальну доступність для користувачів з обмеженими можливостями, включаючи тих, хто не володіє мовою оригіналу контенту. Це дозволяє людям з обмеженими мовними навичками отримувати доступ до інформації на їхній рідній мові або у формі, зручній для сприйняття.

5. Обмеження застосування рівня AAA

Враховуючи, що виконання вимог рівня AAA може вимагати значних технічних зусиль і часу, не всі веб-сайти повинні прагнути до цього рівня доступності. Більш того, у деяких випадках реалізація всіх критеріїв рівня AAA може ускладнити інтерфейс і знизити зручність користування. Тому стандарт WCAG рекомендує застосовувати рівень AAA лише в окремих випадках, тоді як рівень AA вважається оптимальним для більшості сайтів.

Таким чином, розподіл вимог WCAG на три рівні відповідності дозволяє визначити, які саме заходи доступності необхідно впроваджувати залежно від потреб і специфіки користувачів. Рівень А визначає базову мінімальну доступність, рівень AA покращує загальний доступ для більшості людей, а рівень AAA надає можливості для максимального доступу всіх користувачів. Важливо, що всі ці рівні не є взаємовиключними, і для досягнення високого рівня доступності рекомендується поєднувати вимоги всіх трьох рівнів.

2.4. Міжнародна та українська нормативно-правова база щодо доступності

Веб-доступність є важливою частиною забезпечення рівних можливостей для людей з обмеженими можливостями. Для цього в багатьох країнах були розроблені нормативно-правові акти та міжнародні стандарти, що регулюють питання

доступності веб-сайтів і онлайн-ресурсів. Важливо розуміти, як ці нормативи застосовуються в міжнародному контексті та в Україні. У цьому підрозділі ми розглянемо основні міжнародні стандарти та нормативно-правові акти, а також їх імплементацію в Україні.

Міжнародні стандарти та ініціативи з веб-доступності

1. Веб-контент та стандарт WCAG

Міжнародною основою для визначення стандартів веб-доступності є Web Content Accessibility Guidelines (WCAG), або Керівництво з доступності веб-контенту. WCAG було розроблено World Wide Web Consortium (W3C) — організацією, що відповідає за стандарти веб-розробки. Вони надають всесвітньо визнані вимоги до доступності контенту веб-сайтів та веб-додатків для осіб з обмеженими можливостями.[3]

WCAG 2.1, який є останньою версією стандарту, є основним документом, що визначає вимоги для забезпечення доступності веб-контенту. Це документ, який застосовує принципи доступності до таких категорій, як текст, графіка, мультимедійний контент, а також навігація та взаємодія з користувачем. Ключові принципи WCAG 2.1 — це Перцептуальність (Perceivable), Операбельність (Operable), Зрозумілість (Understandable) та Надійність (Robust), які ми розглянули раніше.

2. Декларація прав людини та доступність

Веб-доступність також є частиною міжнародних прав людини. У Декларації прав людини ООН та інших міжнародних угодах, таких як Конвенція про права осіб з інвалідністю, підкреслюється важливість надання рівного доступу до інформації та комунікаційних технологій для всіх осіб без винятку. Ці міжнародні документи накладають на країни зобов'язання щодо забезпечення доступу до веб-контенту для людей з обмеженими можливостями.[10]

3. Європейський стандарт EN 301 549

В Європейському Союзі прийнято EN 301 549, що є стандартом для

доступності ІТ-продуктів та послуг, у тому числі веб-контенту, для осіб з інвалідністю. Цей стандарт застосовується до всіх веб-сайтів, веб-додатків, а також мобільних застосунків та інших цифрових продуктів, що постачаються або використовуються в ЄС. Стандарт EN 301 549 базується на WCAG і визначає технічні вимоги для розробників для забезпечення доступності веб-продуктів.[11, с. 12]

4. Американський Закон ADA (Americans with Disabilities Act)

У США важливим нормативним документом є **Americans with Disabilities Act (ADA)**, який гарантує рівні права для осіб з інвалідністю, у тому числі в цифровому середовищі. Хоча цей закон не вимагає прямого виконання стандарту WCAG, він зобов'язує бізнеси та організації забезпечити доступність своїх веб-сайтів та цифрових послуг для осіб з обмеженими можливостями. В американській юридичній практиці часто використовується інтерпретація вимог ADA через призму WCAG.[12]

5. Австралійський стандарт WCAG 2.0

В Австралії розроблено власну інтерпретацію стандарту WCAG 2.0, зокрема для веб-розробників і урядових установ. Вона підтримує вимоги WCAG 2.0 і є частиною національних політик щодо веб-доступності, таких як Web Content Accessibility Guidelines 2.0 (WCAG 2.0) Australia.[13]

Нормативно-правова база доступності в Україні

1. Закон України "Про основи соціальної захищеності осіб з інвалідністю в Україні"

В Україні основним законодавчим актом, що регулює права осіб з обмеженими можливостями, є Закон України "Про основи соціальної захищеності осіб з інвалідністю в Україні". Він визначає принципи інклюзії, рівності можливостей для осіб з інвалідністю, що також включає доступність інформаційних ресурсів, зокрема веб-сайтів.[14]

Цей закон вимагає, щоб веб-сайти державних органів та інших установ були доступні для людей з інвалідністю, що відповідає вимогам міжнародних

стандартів доступності. Важливим аспектом є забезпечення доступу до публічної інформації через цифрові технології.

2. Постанова Кабінету Міністрів України № 1088 від 2018 року

Важливим нормативним актом є Постанова Кабінету Міністрів України № 1088 від 2018 року, яка регламентує вимоги до доступності веб-сайтів та мобільних додатків для людей з обмеженими можливостями. Постанова визначає, що веб-сайти та інші цифрові ресурси, які належать органам державної влади та місцевого самоврядування, повинні відповідати міжнародним стандартам доступності, зокрема стандарту WCAG 2.0.[15]

Це також охоплює вимоги до доступності інформації, наданої через веб-сайти, та до забезпечення повноцінної взаємодії з онлайн-сервісами для осіб з інвалідністю.

3. Національні стандарти доступності веб-сайтів

Україна також прийняла національні стандарти доступності для веб-сайтів, що є адаптованими до стандарту WCAG 2.0. Ці стандарти застосовуються до державних органів та організацій, а також до підприємств, які надають публічні послуги через Інтернет.[16, с. 3]

4. Закон України "Про інформацію"

Важливим актом є також Закон України "Про інформацію", який зобов'язує державні органи та організації забезпечити доступність публічної інформації, зокрема в цифровому форматі. Це включає забезпечення доступу до інформації для людей з інвалідністю, а також впровадження сучасних технологій, які підтримують веб-доступність.[17]

5. Закон України "Про електронні комунікації"

Закон "Про електронні комунікації" 2020 року також містить положення, які сприяють забезпеченню доступності веб-сайтів для осіб з інвалідністю, встановлюючи вимоги до надання електронних послуг в Інтернеті, які повинні відповідати міжнародним стандартам доступності.[18]

Виконання нормативних вимог в Україні

Попри наявність нормативно-правових актів, що регулюють доступність веб-сайтів в Україні, на практиці реалізація цих стандартів все ще стикається з певними труднощами. В Україні є потреба в посиленні контролю за дотриманням цих норм, а також у підвищенні обізнаності серед розробників веб-сайтів і підприємців щодо важливості забезпечення доступності.

З одного боку, існує поступове впровадження стандартів WCAG в державні установи та організації, що працюють з публічною інформацією. Однак для досягнення справжньої інклюзивності потрібно посилювати підтримку від уряду та громадських організацій у напрямку забезпечення доступності на всіх рівнях.

2.5. Приклади порушень доступності на практиці

Проблеми з контрастом тексту та фону

Однією з найпоширеніших і водночас найбільш недооцінених проблем доступності є недостатній контраст між текстом і фоном. Багато дизайнерів, орієнтуючись на візуальну естетику, обирають кольорові рішення, які виглядають привабливо для більшості користувачів. Однак такі рішення можуть бути катастрофічно незручними або взагалі непридатними для людей з порушеннями зору, включаючи кольоросліпоту, знижену контрастну чутливість або загальне зниження гостроти зору.

Контрастність — це один з найважливіших параметрів сприйняття інформації. Якщо текст «зливається» з фоном, або відтінки надто схожі, користувачеві стає важко або неможливо прочитати інформацію, навіть якщо вона технічно присутня на сторінці.

Визначення поняття контрасту та його значення

Контраст у контексті веб-дизайну — це різниця у яскравості (світлості) між кольором тексту і фону, на якому цей текст розміщується. Ця різниця вимірюється у

вигляді коефіцієнта, що обчислюється відповідно до формули, визначеної стандартом WCAG.

Для нормального тексту WCAG 2.1 встановлює мінімальний рівень контрасту — 4.5:1. Це означає, що світлість тексту і фону повинні відрізнятися настільки, щоб люди з порушеннями зору могли розрізняти літери без труднощів. Для великого тексту (розміром щонайменше 18 пт або 14 пт жирним) вимога м'якша — 3:1.

Значення контрасту особливо важливе для:

- літніх людей;
- користувачів з віковою дегенерацією зору;
- людей з тимчасовими порушеннями (втома очей, бликуючий екран, робота на сонці).

Типові приклади порушення контрасту

Приклад 1: Світло-сірий текст на білому фоні.

Дизайн виглядає мінімалістично і «чисто», однак користувачі з порушенням зору або при яскравому освітленні можуть взагалі не побачити текст.

Example Example Example

Рис. 2.3. Приклад порушення контрасту. Світло сірий текст на білому фоні.

Джерело: розроблено автором

Приклад 2: Жовтий текст на білому або світло-сірому фоні.

Це рішення часто зустрічається в молодіжних та креативних сайтах, але кольори настільки близькі за світлістю, що читання стає дуже ускладненим.

Example Example Example

Рис. 2.4. Приклад порушення контрасту. Жовтий текст на білому фоні.

Джерело: розроблено автором

Приклад 3: Темно-синій текст на темно-зеленому фоні.

Обидва кольори можуть мати схожий рівень яскравості, і хоча кольорове розрізнення може бути можливим для здорових очей, для користувачів з дальтонізмом таке поєднання буде «злитим».



Рис. 2.5. Приклад порушення контрасту. Темно-синій текст на темно-зеленому фоні.

Джерело: розроблено автором

Наслідки низького контрасту

- **Погіршення читабельності.** Людина мусить докладати зусиль, щоби розпізнати текст.
- **Зниження інклюзивності.** Люди з порушеннями зору можуть взагалі не мати доступу до текстової інформації.
- **Порушення стандарту WCAG.** Такі сайти не відповідають рівням А чи АА, що може призвести до юридичних проблем у країнах, де доступність є обов'язковою.

Як перевіряти контрастність

Існують численні інструменти для перевірки контрасту кольорів:

- **WebAIM Color Contrast Checker**

- **Contrast Ratio (by Lea Verou)**
- **Google Lighthouse accessibility audit**

Вони дозволяють швидко перевірити комбінацію кольорів і отримати числове значення контрасту, а також вказують, чи відповідає дана пара кольорів стандарту WCAG на рівнях A, AA або AAA.

Рекомендації та хороші практики

1. **Використовуйте висококонтрастні кольори** для тексту, особливо в навігації, заголовках, формах і кнопках.
2. **Тестуйте колірні пари перед релізом**, особливо якщо використовуються нестандартні палітри або брендові кольори.
3. **Забезпечте адаптивність дизайну**, щоб контраст не зникав при масштабуванні або зміні яскравості екрана.

Відсутність доступних форм і елементів керування

Однією з найсерйозніших, але при цьому найпоширеніших проблем, пов'язаних з веб-доступністю, є саме недоступність форм та елементів керування, що застосовуються на веб-сайтах. Форми — це невід'ємна складова більшості веб-інтерфейсів. Саме через форми відбувається взаємодія користувача із системою: заповнення профілю, авторизація, пошук, фільтрація, зворотний зв'язок, бронювання тощо. Проте для людей з інвалідністю, особливо з порушенням зору, моторики, когнітивними розладами або неврологічними особливостями, саме ці елементи можуть стати непереборною перешкодою, якщо вони реалізовані з порушенням принципів доступності.

Багато розробників нехтують елементарними вимогами до оформлення форм: часто немає зв'язку між підписом та полем, не надаються інструкції або повідомлення про помилки, або ж ці повідомлення візуально виражені лише кольором. Це робить взаємодію з формою складною або навіть неможливою для користувачів з певними особливостями.

Недоступні мітки полів (label)

Однією з найпоширеніших проблем є відсутність правильно зв'язаних міток (<label>) із відповідними полями введення. Скрінрідери, що озвучують вміст сторінки для людей із порушенням зору, орієнтуються саме на ці зв'язки. Коли елемент <label> не асоційований із відповідним полем введення через атрибут `for`, користувач не отримує інформації про те, що саме потрібно ввести. У результаті — незрозумілий інтерфейс, навіть якщо він візуально виглядає цілком логічним.

Для прикладу розглянемо фрагмент HTML-коду:

Лістинг 2.1. Невдалий приклад реалізації текстового поля без зв'язаної мітки

```
<input type="text" id="name">
```

У цьому випадку поле не має зв'язаної мітки, тому скрінрідер не озвучує, що саме потрібно ввести.

Лістинг 2.2. Вдалий приклад реалізації текстового поля з пов'язаною міткою

```
<label for="name">Ім'я:</label>  
  
<input type="text" id="name">
```

Тут мітка пов'язана з полем введення за допомогою атрибута `for`, що дозволяє скрінрідеру коректно озвучити інструкцію користувачеві.

Альтернативно, для спрощення розмітки можна використовувати вкладення елемента <input> безпосередньо в мітку:

Лістинг 2.3. Вдалий приклад: обгортання поля вводу всередині мітки <label>

```
<label>Ім'я: <input type="text" name="name"></label>
```

Обидва варіанти забезпечують необхідну доступність і зручність для користувачів із порушенням зору.

Недоступні повідомлення про помилки

Багато форм не передбачають доступної системи повідомлень про помилки. Наприклад, користувач натискає кнопку "Надіслати", залишивши обов'язкове поле порожнім. Система просто підсвічує його червоним, не вказуючи, що сталося. Це створює значні бар'єри для користувачів з вадами зору або з дальтонізмом — вони можуть просто не побачити колірне виділення, не зрозуміти, де саме помилка і що потрібно виправити.

У хороших практиках повідомлення про помилки подаються у вигляді:

- текстового повідомлення (наприклад, *"Поле Email обов'язкове для заповнення"*),
- розміщення повідомлення поруч із відповідним полем,
- використання `aria-describedby` для озвучування повідомлення скрінрідером,
- зміна статусу елемента через `aria-invalid="true"`.

Така реалізація значно покращує зручність користування для всіх категорій користувачів, включаючи людей без інвалідності.

Недоступність елементів керування з клавіатури

Ще однією суттєвою проблемою є недоступність форм для навігації з клавіатури. Для багатьох користувачів основним засобом взаємодії із сайтом є саме клавіатура: наприклад, люди з ураженням моторики рук не використовують мишу, а орієнтуються виключно на клавіатурні комбінації або спеціальні перемикачі. У таких

випадках всі інтерактивні елементи (поля, перемикачі, випадаючі списки, кнопки) мають бути доступними через клавішу Tab та реагувати на Enter, Space чи відповідні комбінації.

Форми, які реалізовані лише з орієнтацією на мишу, повністю виключають таких користувачів із процесу. Наприклад, стилізована кнопка надсилання без семантичного HTML-елемента `<button>` може виглядати ефектно, але не бути доступною для активації з клавіатури.

Важливість у контексті бібліотечної системи

У випадку системи управління бібліотекою з онлайн-каталогом та резервуванням книг наявність доступних форм є критично важливою. Адже користувачам необхідно:

- виконати пошук за автором, назвою або темою,
- зареєструватися або увійти до особистого кабінету,
- зробити запит на бронювання книги,

Кожен з цих сценаріїв базується на заповненні форми. Якщо ці форми не відповідають стандартам доступності, то значна частина користувачів буде просто виключена з користування електронною бібліотечною системою, що суперечить як принципам WCAG, так і принципам інклюзивності в освіті та доступу до культури.

Складність навігації для людей з обмеженими можливостями

Навігація — це фундаментальний елемент будь-якого веб-ресурсу. Вона визначає логіку пересування користувача між сторінками, розділами, елементами інтерфейсу, впливає на швидкість та ефективність взаємодії з контентом. Для користувачів без обмежень можливостей навігаційні труднощі можуть бути лише незначною незручністю. Однак для людей з інвалідністю будь-яке ускладнення навігації може стати вирішальним бар'єром, що унеможливорює подальше використання веб-сайту.

Визначення навігаційної доступності

Навігаційна доступність — це здатність користувача швидко, зрозуміло і безперешкодно орієнтуватися у структурі веб-сайту за допомогою доступних засобів введення (клавіатура, скрінрідер тощо). Сюди входять такі аспекти:

- зрозуміле меню та логічна структура розділів;
- правильне позначення активних елементів (посилань, кнопок);
- можливість легко визначити, де саме користувач знаходиться на сайті;
- передбачувана поведінка при переході між сторінками;
- доступність навігації без миші.

Бар'єри навігації, що виникають на практиці

1. Відсутність або складність використання клавіатурної навігації

Багато користувачів, особливо з моторними або нейром'язовими розладами, не мають можливості користуватися мишею. Для них навігація за допомогою клавіші Tab, стрілок, Enter або Esc є основним способом взаємодії з веб-сайтом. Утім, якщо елементи інтерфейсу не мають чіткого фокусу або не дозволяють пересування через клавіатуру, доступ до потрібних розділів стає практично неможливим.

Особливо часто ця проблема трапляється в сучасних адаптивних сайтах, де меню або інші блоки відкриваються через JavaScript без забезпечення правильних ARIA-атрибутів (aria-expanded, aria-controls тощо), що інформують допоміжні технології про зміни стану елементів.

2. Відсутність візуального фокусу

Коли користувач пересувається між елементами за допомогою клавіатури, кожен елемент повинен чітко візуально вказувати, що він активний. Це реалізується через стилізацію фокусу (:focus). Якщо дизайн сайту прибирає або маскує стандартне обведення фокусу (outline), користувач не бачить, де саме він

знаходиться, і втрачає орієнтацію. Це може призвести до помилок, дезорієнтації або повного виходу з процесу навігації.

3. Недостатньо зрозуміла або перевантажена структура меню

Меню — це один із ключових навігаційних компонентів. Воно має бути інтуїтивно зрозумілим, лаконічним, ієрархічно впорядкованим. На жаль, багато сайтів містять складні вкладені меню, в яких важко орієнтуватися. Якщо при цьому не передбачено логічних груп, назви пунктів не інформативні або повторюються — користувач, особливо з когнітивними порушеннями або порушенням уваги, просто «губиться» у навігації.

4. Непередбачувана поведінка елементів

Навігація має бути послідовною та передбачуваною. Якщо після кліку на посилання або кнопку користувач очікує, що відкриється сторінка, а натомість з'являється спливаюче вікно, модальне вікно або відбувається перехід в новій вкладці без попередження, це може викликати плутанину або дезорієнтацію. Для користувачів з порушенням зору така поведінка ще гірша, адже вони не бачать візуальних підказок і можуть не помітити, що з'явилося нове вікно чи вкладка.

5. Відсутність “aria landmarks” та семантики HTML5

HTML5 надає чудовий набір семантичних тегів (<nav>, <header>, <main>, <aside>, <footer>), які допомагають скрінрідерам розпізнавати основні зони на сторінці. Також існують ARIA-регіони (role="navigation", role="main"), які дублюють або доповнюють HTML5. Якщо ці маркери відсутні, скрінрідер не може швидко перескочити до потрібного розділу — користувач змушений повільно “прослуховувати” всю сторінку від початку.

Висновки до розділу 2

У цьому розділі було розглянуто ключові аспекти веб-доступності як важливого компоненту сучасного цифрового середовища. Визначено поняття

веб-доступності, яке полягає у створенні умов, за яких всі користувачі, включно з особами з інвалідністю, мають рівний доступ до інформації та функціональності веб-ресурсів. Основою стандарту веб-доступності є рекомендації WCAG 2.1, які формують чітку структуру вимог, поділених за принципами POUR (сприйнятність, керуваність, зрозумілість, надійність).

Стандарти WCAG передбачають три рівні відповідності — А, АА та ААА, що дозволяє поступово підвищувати рівень доступності залежно від потреб і можливостей розробника чи організації. Важливу роль у впровадженні веб-доступності відіграють нормативно-правові документи, як міжнародні, так і українські, які закріплюють вимоги до інклюзивного дизайну веб-ресурсів на законодавчому рівні.

Окрім теоретичного аналізу, у розділі наведено приклади реальних порушень доступності, які демонструють типові помилки під час створення сайтів і підкреслюють необхідність дотримання стандартів WCAG. Отже, інформаційне забезпечення у сфері веб-доступності є критично важливим етапом для розуміння і впровадження ефективних рішень, що сприятимуть створенню інклюзивного цифрового простору.

РОЗДІЛ 3.

ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1. Архітектура системи та загальний підхід

Архітектура системи (стек технологій, Vue.js тощо)

Система управління бібліотекою з онлайн-каталогом та функціоналом бронювання книг реалізована як сучасний **веб-застосунок**, що базується на передових принципах односторінкових додатків (SPA – *Single Page Application*). Такий підхід забезпечує максимально динамічну взаємодію користувача з інтерфейсом, коли нові сторінки не потребують повного перезавантаження браузера, що особливо важливо для підвищення зручності користування й швидкості реагування системи.

Технологічна основа проєкту

Основним фреймворком, на якому побудовано клієнтську частину веб-застосунку, є **Vue.js** версії 3. Це одна з найпопулярніших JavaScript-бібліотек, орієнтована на створення реактивних інтерфейсів з відкритою структурою. У порівнянні з попередніми версіями, **Vue 3** підтримує новий Composition API, що дає змогу більш гнучко організовувати логіку компонентів, повторно використовувати код, підвищувати читаємість проєкту та зменшувати дублювання функціоналу.

Запуск і конфігурація проєкту здійснювалась через Vue CLI (Command Line Interface) — офіційний інструмент Vue-екосистеми, який дозволяє швидко створювати, будувати та масштабувати проєкти. CLI автоматизує багато процесів, включаючи налаштування транспіляції, підключення модулів, лінтерів, збірників та інших інструментів, які є базовими у сучасній веб-розробці.

Кодова база системи чітко організована у папці `src/`, з поділом на логічні блоки:

- `components/` – основна директорія для зберігання компонентів. Компоненти структуровано у підтексти `admin`, `auth`, `main`, `profile`, що дозволяє легко орієнтуватись в архітектурі проекту.
- `layouts/` – шаблони, які використовуються для організації зовнішнього вигляду сторінок. Наприклад, `MainLayout.vue` включає хедер, футер та сайдбар.
- `router/` – реалізація навігації між сторінками застосунку, налаштована з використанням `Vue Router`.
- `store/` – централізований стан застосунку, реалізований через `Vuex`.

Також у кореневій директорії присутні конфігураційні файли (`main.js`, `App.vue`, `.gitignore`, `babel.config.js`) та API-обгортка (`api.js`), яка відповідає за взаємодію з бекендом.

Огляд функціональних компонентів

Інтерфейс застосунку складається з великої кількості компонентів, які згруповані за логікою використання:

- **Основні компоненти інтерфейсу (`main/`):** охоплюють головну сторінку (`HomePage.vue`), блок з правилами користування (`RulesComp.vue`), список книг (`BookList.vue`), карусель із цитатами (`QuoteOfTheDay.vue`), а також інформацію про проєкт (`AboutUs.vue`).
- **Інтерфейсні блоки (`HeaderLibrary.vue`, `FooterLibrary.vue`, `SidebarLibrary.vue`):** організовують глобальний UI, що повторюється на більшості сторінок. Усі ці елементи можна легко адаптувати для доступності.

Використані бібліотеки та залежності

У проєкті реалізовано використання таких зовнішніх бібліотек і пакетів:

- `vue-router` — маршрутизація між сторінками без перезавантаження.
- `vuex` — централізоване управління станом застосунку.
- `sweetalert2` — створення діалогових вікон з врахуванням доступності.

- swiper, @splidejs/vue-splide — реалізація каруселей для елементів головної сторінки.

Потенціал для реалізації доступності

Оскільки застосунок побудований без використання сторонніх UI-фреймворків (як-от Vuetify чи BootstrapVue), розробник має повний контроль над розміткою HTML, поведінкою JavaScript та атрибутами доступності (aria-*, role, tabindex тощо). Це дозволяє максимально адаптувати інтерфейс під вимоги стандарту **WCAG 2.1**, уникаючи конфліктів зі стилями та поведінкою за замовчуванням, що часто присутні в таких фреймворках.

Це також створює підґрунтя для подальшої імплементації кастомних компонентів із високим рівнем доступності — наприклад, доступних модальних вікон, навігаційних меню з підтримкою клавіатурного фокусу, читабельних форм для реєстрації тощо.

Загальний підхід до імплементації доступності

На початкових етапах проєктування веб-застосунку системи управління бібліотекою декларувалося прагнення враховувати вимоги веб-доступності згідно зі стандартом WCAG 2.1. Їхня реалізація у першій версії інтерфейсу була поверхневою. Інтерфейс не підтримував навігацію з клавіатури, не мав чіткої семантичної структури HTML, не містив ARIA-атрибутів тощо — усе це фактично робило систему недоступною для користувачів з інвалідністю.

У розробці було прийнято інтегрований підхід до імплементації доступності, який охоплює як архітектурні рішення, так і практики безпосередньої розмітки та стилізації інтерфейсу. Основна увага приділялася дотриманню чотирьох ключових принципів WCAG: Сприйнятність (Perceivable), Керованість (Operable), Зрозумілість (Understandable) та Надійність (Robust).

Основні етапи та принципи реалізації

1. Проектування структури з урахуванням доступності

На початковому етапі було розроблено логічну структуру застосунку, що базується на семантичній верстці та правильному використанні тегів HTML5. Компоненти розроблялися так, щоб інтерфейс був зрозумілим при використанні лише клавіатури або скрінрідера. Було впроваджено чітку ієрархію заголовків (<h1>–<h4>), логічний порядок переходу між елементами (tabindex), розмітку форм із відповідними <label>.

2. Дотримання принципу мобільності та масштабованості

Оскільки застосунок призначено для широкого кола користувачів, було важливо забезпечити адаптивність дизайну. Компоненти автоматично масштабуються відповідно до розміру екрана, що дозволяє користувачам працювати із сайтом на мобільних пристроях або при зміненому масштабі у браузері. Це також критично для користувачів із зниженим зором.

3. Забезпечення доступності з клавіатури

Користувачі, які не можуть використовувати мишу (наприклад, через порушення моторики), мають можливість повноцінно користуватися сайтом лише за допомогою клавіатури. Було впроваджено гарячі клавіші (Alt+1, Alt+2, тощо) для швидкої навігації між основними розділами. Всі інтерактивні елементи отримують фокус у логічному порядку, реалізовано візуальне відображення фокусування.

4. Дотримання правил кольорового контрасту та видимості

Колірна палітра, обрана для застосунку, відповідає мінімальним контрастним співвідношенням, визначеним WCAG 2.1 для рівня AA. Текст має належний контраст з фоном, усі фокусні та активні стани чітко позначені. Для зорової орієнтації також використано додаткові візуальні маркери, наприклад, підсвічування кнопок при фокусі.

5. Використання ARIA-атрибутів

У місцях, де базової HTML-розмітки недостатньо для опису поведінки елементів, використовуються ARIA (Accessible Rich Internet Applications) атрибути, зокрема:

- `aria-label`, `aria-labelledby` — для уточнення призначення елементів,
- `aria-required`, `aria-invalid` — для валідації форм,
- `role="navigation"`, `role="main"` — для структурування основних областей застосунку.

ARIA-атрибути допомагають скрінрідерам правильно інтерпретувати призначення кожного елемента.

6. Тестування доступності на кожному етапі

Після реалізації кожного компонента проводилася перевірка за допомогою інструменту Lighthouse. Він дозволяє оперативно виявити типові помилки доступності (наприклад, відсутність альтернативного тексту, неправильну вкладеність заголовків, низький контраст) та усунути їх ще до фінального етапу тестування.

Переваги обраного підходу

Завдяки системному впровадженню доступності на всіх етапах розробки, вдалося досягти наступного:

- **Інклюзивності** — застосунок придатний до використання широким колом користувачів, включаючи людей із порушенням зору, моторики, когнітивними особливостями.
- **Універсального дизайну** — інтерфейс зрозумілий та зручний навіть для користувачів, які не мають досвіду взаємодії з веб-ресурсами.
- **Якісної архітектурної основи** — впроваджені рішення легко масштабуються, що дозволяє в майбутньому додавати нові функції без шкоди для доступності.

У цілому, обраний підхід базується на ідеї «accessibility by design» — тобто доступність не як додатковий модуль, а як невід’ємна частина проєктування та розробки системи. Завдяки цьому забезпечується відповідність ключовим вимогам WCAG 2.1 рівня AA, що є рекомендованим стандартом для академічних і державних вебресурсів.

3.2. Реалізація принципів WCAG у системі

Реалізація принципу «Сприйнятність» (Perceivable)

Принцип «Сприйнятність» (Perceivable) є першим із чотирьох основних принципів WCAG 2.1. Його суть полягає в тому, що користувач повинен мати змогу сприйняти інформацію і компоненти інтерфейсу незалежно від своїх фізичних можливостей. Іншими словами, нічого в інтерфейсі не повинно бути «невидимим» або недоступним для сприйняття — незалежно від того, користувач взаємодіє з ним через зір, слух або допоміжні технології.

У розробленій системі управління бібліотекою було реалізовано ряд заходів, що відповідають вимогам принципу «Сприйнятність» відповідно до критеріїв успішності рівня AA.

1. Текстові альтернативи для мультимедійних чи графічних елементів інтерфейсу

Всі зображення, які використовуються в системі (зокрема обкладинки книг), мають атрибут `alt`, що описують зміст зображення. Це дозволяє користувачам, які використовують скрінрідери, отримати ту саму інформацію у текстовому вигляді. Наприклад:

Лістинг 3.1. Приклад використання атрибута `alt` для зображення автора книги

```

```

Для декоративних елементів, які не несуть змістового навантаження, використано порожній атрибут `alt=""`, що запобігає зайвому озвучуванню скрінрідерами.

2. Семантична структура і заголовки

Усі сторінки бібліотечної системи мають логічну структуру з чіткою ієрархією заголовків (`<h1>`, `<h2>`, `<h3>`) і змістових блоків (`<main>`, `<nav>`, `<aside>`, `<footer>`). Це дозволяє користувачам легко орієнтуватися у змісті, переходити між секціями, а також дає змогу асистивним технологіям правильно інтерпретувати сторінку. Наприклад:

Лістинг 3.2. Приклад використання заголовків для логічної структури сторінки

```
<h1>Онлайн-каталог бібліотеки</h1>  
  
<h2>Популярні книги</h2>  
  
<h2>Новинки</h2>
```

Це також допомагає формувати логічний шлях взаємодії і відповідає рекомендаціям WCAG щодо структурованості контенту.

3. Колірний контраст і стилі

На головній сторінці застосунку реалізовано дизайн з урахуванням вимог до контрастності згідно з WCAG 2.1 рівня AA. Для звичайного тексту забезпечено контрастне співвідношення не менше 4.5:1, а для великого тексту — не менше 3:1. Наприклад, у секції «Цитата дня» на градієнтному тлі використано білий жирний

шрифт великого розміру, що дозволяє легко сприймати зміст навіть користувачам зі зниженим зором.

Для основного інтерфейсу (меню, фільтри, кнопки) використано темний шрифт на світлому фоні, що також відповідає вимогам доступності. Активні елементи мають візуально помітні стилі фокусування — при наведенні або отриманні фокусу застосовуються зміни кольору, тінь, або обводка, що дозволяє легко ідентифікувати ці елементи. Наприклад, кнопки фільтрації чітко виділяються зміною обрамлення, а поля вводу мають світлу рамку з додатковим фокус-ефектом.

Ці рішення роблять інтерфейс зручним для людей із порушеннями зору або кольоросприйняття, а також покращують загальну юзабіліті системи.

4. Масштабування та адаптивність

Інтерфейс було реалізовано з урахуванням вимог до масштабування. При збільшенні масштабу до 200% функціональність залишається повністю доступною. Деякі елементи інтерфейсу (зокрема сайдбар) автоматично адаптуються до компактнішого вигляду, однак горизонтального прокручування не виникає, що відповідає стандарту WCAG 2.1. Це особливо важливо для користувачів із низьким зором. Адаптивна верстка забезпечується за допомогою медіа-запитів CSS та використання гнучких блоків (flex, grid).

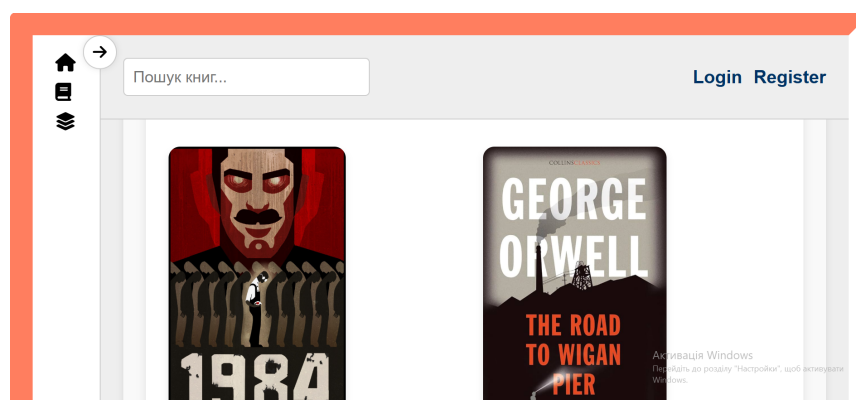


Рис. 3.1. Вигляд веб-застосунку при масштабі 200%

Джерело: розроблено автором

5. Використання розмітки форм і підказок

Усі поля введення у формах мають відповідні мітки (<label>), які логічно пов'язані з полем (for="input-id"). Також реалізовано підказки (placeholder) і повідомлення про помилки з чіткими текстовими описами, доступними для скрінрідерів:

Лістинг 3.3. Приклад пов'язаної мітки та поля введення з підказкою

```
<label for="search">Пошук книги</label>  
  
<input id="search" type="text" placeholder="Введіть назву  
або автора">
```

7. Інформування про зміну контенту

У застосунку не використовуються автоматичні оновлення контенту, які можуть бути непомітні для скрінрідерів. У випадках, де контент може динамічно змінюватися (наприклад, при фільтрації книг), можна використати ARIA-атрибути, такі як aria-live="polite" для забезпечення повідомлення користувачам про зміну.

8. Альтернативи для мультимедійного контенту

У разі розширення системи в майбутньому із впровадженням відео- або аудіо-контенту передбачено реалізацію субтитрів або транскриптів, що також є вимогою WCAG для принципу сприйнятності. На поточному етапі такий контент не застосовується.

У реалізованих функціях інтерфейсу враховано ключові вимоги принципу «Сприйнятність» відповідно до стандарту WCAG 2.1 рівня AA. Особливу увагу було приділено створенню інтерфейсу, який буде доступним для сприйняття всіма користувачами, включаючи осіб із вадами зору, кольоросприйняття або користувачів, які застосовують допоміжні технології. Завдяки використанню семантичної

розмітки, забезпеченню контрасту, адаптивного дизайну та текстових альтернатив, застосунок є надійною основою для подальшої інклюзивної розробки.

Реалізація принципу «Керованість» (Operable)

Принцип «Керованість» (Operable) означає, що всі функції інтерфейсу мають бути доступні за допомогою клавіатури та легко керовані користувачами з різними можливостями. Це включає підтримку навігації без миші, наявність чіткої фокус-індикації, логічну структуру керованих елементів та уникнення елементів, які викликають судоми, наприклад блимаючого контенту.

У реалізованій бібліотечній системі ці вимоги були враховані на кількох рівнях:

1. Клавіатурна навігація (Keyboard Accessibility)

Користувачі можуть переміщатися між основними розділами сайту за допомогою клавіш швидкого доступу. Це забезпечено у файлі App.vue, де були реалізовані комбінації клавіш Alt + цифра, які переміщують користувача до конкретної сторінки:

Лістинг 3.4. Приклад реалізації клавіш швидкого доступу для навігації між розділами сайту

```
mounted() {  
  window.addEventListener("keydown", (e) => {  
    if (e.altKey) {  
      switch (e.key) {  
        case "1":
```

```

        this.$router.push("/");

        break;

    case "2":

        this.$router.push("/booklist");

        break;

    case "3":

        this.$router.push("/login");

        break;

    }

}

});

}

```

Це дозволяє швидко переміщатися між головною сторінкою, каталогом книг і формою авторизації без використання миші, що особливо важливо для користувачів із порушеннями моторики.

2. Фокус і візуальні підказки

Кожен інтерактивний елемент (наприклад, картка книги) має `tabindex="0"` — це забезпечує можливість навігації за допомогою клавіші Tab. Крім того, при фокусі на елементі застосовується візуальний індикатор фокусу (рамка і масштабування):

Лістинг 3.5. Приклад забезпечення доступності через tabindex та візуальний індикатор фокусу

```
<div
  v-for="book in books"
  :key="book.bookid"
  tabindex="0"
  class="book-item"
  @keydown.enter="goToBookDetails(book.bookid)"
>

.book-item:focus {
  outline: none;
  box-shadow: 0 0 15px 5px rgba(0, 123, 255, 0.6);
  transform: scale(1.02);
  transition: all 0.2s ease-in-out;
  z-index: 0;
}
```

Це дозволяє користувачеві бачити, який елемент наразі у фокусі, і взаємодіяти з ним клавішею Enter. Наприклад, при натисканні Enter на картці книги відбувається перехід до її сторінки деталей.

3. Клавiшні обробники та альтернативна активація

Важливою частиною керованості є можливість активувати елементи не лише кліком миші, а й через клавіатуру. Це було реалізовано в обробнику події клавiші Enter:

Лістинг 3.6. Приклад обробки клавiші Enter для активації елемента через клавіатуру

```
@keydown.enter="goToBookDetails(book.bookid)"
```

Це гарантує, що навіть якщо користувач не може клікнути мишкою по обкладинці книги, він зможе відкрити її опис за допомогою клавіатури.

4. Фільтри й селектори — доступ до форм

Усі елементи керування, зокрема селектори для фільтрації (жанр, автор, видавництво), мають відповідні підписи (<label>), зв'язані через атрибути for та id, що забезпечує доступність для скрінрідерів і клавіатурної навігації:

Лістинг 3.7. Приклад зв'язування мітки з елементом select для забезпечення доступності

```
<label for="authorid">Автор:</label>

<select      id="authorid"      v-model="filters.authorid"
@change="fetchBooks">

    ...

</select>
```

5. Візуальна передбачуваність і уникнення пасток фокусу

Компоненти розміщені в логічному порядку, що дозволяє інтуїтивно переходити від одного елемента до іншого. Уся навігація передбачувана: фокус

переходить по селекторам, а потім — до списку книг. Жодного прихованого або недоступного фокусного елемента в системі не реалізовано.

6. Перехід між сторінками — плавні анімації та контрольоване з'явлення

Компоненти `transition` із CSS-анімаціями знижують когнітивне навантаження й допомагають користувачу орієнтуватись у зміні контенту:

Лістинг 3.8. Приклад реалізації плавних анімацій переходів за допомогою Vue `<transition>`

```
<transition  
  name="fade"  
  @before-enter="beforeEnter"  
  @enter="enter"  
  @leave="leave"  
>  
  
beforeEnter(el) {  
  el.style.opacity = 0;  
},  
  
enter(el, done) {  
  el.offsetHeight;  
  
  el.style.transition = "opacity 1s";
```

```
el.style.opacity = 1;  
  
done();  
  
},
```

Таким чином, принцип «Керованість» було реалізовано згідно з рекомендаціями WCAG 2.1. Основна увага приділена забезпеченню доступу через клавіатуру, наявності візуального фокусу, підтримці альтернативних шляхів навігації та логічній структурі управління. Це суттєво покращує інклюзивність бібліотечної системи для широкого кола користувачів.

Реалізація принципу «Зрозумілість» (Understandable)

Принцип «Зрозумілість» (Understandable) у стандарті WCAG 2.1 полягає в тому, що інтерфейс, контент і навігація мають бути зрозумілими та передбачуваними. Система повинна працювати логічно та надавати користувачам необхідний зворотний зв'язок. Навіть без досвіду користування, людина має змогу інтуїтивно зрозуміти, як нею користуватись.

У бібліотечній системі реалізовано низку рішень, які забезпечують високий рівень зрозумілості:

1. Інтуїтивний інтерфейс

Інтерфейс побудований за загальнозживаними шаблонами взаємодії: верхнє меню, сайдбар, список книг, авторизація, детальна сторінка книги. Кожна сторінка має чітку структуру та однозначну мету. Це зменшує когнітивне навантаження на користувача.

Приклад: у списку книг (booklist.vue) є фільтри за автором, видавництвом та жанром — ці елементи розташовані перед списком книжок, що логічно і зручно для користувача:

Лістинг 3.9. Приклад логічного розташування фільтрів у списку книг

```
<label for="authorid">Автор:</label>

<select      id="authorid"      v-model="filters.authorid"
@change="fetchBooks">...</select>
```

2. Передбачувана поведінка елементів

Усі інтерактивні елементи поведуться послідовно:

- **Натискання на картку книги** завжди відкриває деталі цієї книги.
- **Натискання Enter** на фокусованому елементі також виконує ту саму дію, що і клік миші.
- Фільтри працюють одразу після вибору — не потребують окремого підтвердження, що зменшує кількість дій з боку користувача.

Лістинг 3.10. Приклад забезпечення передбачуваної поведінки інтерактивних елементів через обробку клавіші Enter

```
@keydown.enter="goToBookDetails(book.bookid)"
```

3. Зрозумілі назви кнопок та елементів

Усі кнопки, посилання та селектори мають зрозумілі назви українською мовою. Наприклад, замість технічних термінів використовуються прості позначення: "Автор", "Видавництво", "Жанр", "Пошук книг", тощо.

4. Семантична розмітка

Використання правильних HTML-елементів сприяє тому, що навіть без візуального інтерфейсу (наприклад, при озвучуванні скрінрідером), користувач розуміє контекст. Наприклад:

Лістинг 3.11. Приклад семантичної розмітки з використанням `<main>`, `<section>`, `<label>`, `<select>`

```
<main>

  <section class="filters">

    <label for="genreid">Жанр:</label>

    <select id="genreid" ...>...</select>

  </section>


  <section class="booklist">

    <div

      v-for="book in books"

      class="book-item"

      tabindex="0"

      @keydown.enter="goToBookDetails(book.bookid)"

    >

      {{ book.bookname }}

    </div>

  </section>

</main>
```

Тут `<main>`, `<section>`, `<label>`, `<select>` виконують не лише візуальну, а й семантичну роль, роблячи структуру сторінки зрозумілою як для людини, так і для технологій допомоги.

5. Мінімізація неочікуваних змін інтерфейсу

Немає спливаючих вікон без попередження або автоматичних редиректів, які можуть збити користувача з пантелику. Перехід між сторінками виконується лише після явної дії користувача (клік або Enter), і завжди веде до очікуваного результату.

6. Мова сторінки та локалізація

У застосунку використано українську мову інтерфейсу. Для підтримки вимог доступності та коректного озвучування контенту екранними читалками, у кореневий елемент `index.html` було додано атрибут `lang="uk"`.

Принцип «Зрозумілість» реалізовано шляхом створення інтуїтивного інтерфейсу, передбачуваної поведінки елементів, семантичної структури сторінок і використання зрозумілої мови. Завдяки цьому система стає більш дружньою до користувачів різного рівня підготовки, включаючи людей з порушеннями сприйняття або новачків.

Реалізація принципу «Надійність» (Robust)

Принцип «Надійність» (Robust) згідно з WCAG 2.1 передбачає, що контент має бути достатньо надійним, щоб його могли правильно інтерпретувати як сучасні, так і майбутні користувацькі агенти — зокрема браузері, допоміжні технології (як-от скрінрідери), мобільні пристрої. Інакше кажучи, веб-інтерфейс повинен бути сумісним із широким спектром технологій.

1. Використання сучасних веб-технологій (Vue.js 3)

Проект побудовано на основі **Vue.js 3** — одного з найпоширеніших і стабільних JavaScript-фреймворків сучасності. Він дотримується стандартів

ECMAScript, має підтримку для рендерингу через Virtual DOM і сумісний із більшістю сучасних браузерів (Chrome, Firefox, Safari, Edge). Завдяки цьому компоненти веб-застосунку надійно працюють навіть у мобільному середовищі.

2. Семантична HTML-розмітка

Для забезпечення коректної роботи з технологіями доступності (зчитувачі екрану, голосові помічники) важливо, щоб сторінка мала семантичну структуру. Це реалізовано за допомогою таких тегів, як `<main>`, `<section>`, `<label>`, `<select>`, `<button>`, `<nav>` тощо.

Лістинг 3.12. Приклад семантичної розмітки для забезпечення доступності

```
<main>

  <section>

    <label for="publisherid">Видавництво:</label>

    <select id="publisherid" v-model="filters.publisherid"
@change="fetchBooks">...</select>

  </section>

</main>
```

Таке використання тегів допомагає скрінрідерам краще інтерпретувати призначення елементів.

3. Коректне використання атрибутів доступності (ARIA, tabindex, alt)

У коді застосовуються стандартизовані атрибути доступності:

- `tabindex="0"` — для забезпечення фокусування на елементах списку книг клавіатурою.

- @keydown.enter — для реалізації дії при натисканні Enter.
- alt — у випадку додавання зображень книг (планується на наступних етапах).
- role="button" і aria-label — у майбутньому можуть бути застосовані для кастомних компонентів.

Лістинг 3.13. Приклад використання атрибутів доступності (tabindex, keyboard events) для інтерактивних елементів

```
<div  
  
  class="book-item"  
  
  tabindex="0"  
  
  @keydown.enter="goToBookDetails(book.bookid)"  
  
>  
  
  {{ book.bookname }}  
  
</div>
```

4. Сумісність з технологіями допомоги (assistive technologies)

Дотримання семантики та використання tabindex гарантує, що скрінрідери можуть оглядати й озвучувати елементи, а користувачі - пересуватись сторінкою без миші. Це підвищує надійність доступу до інтерфейсу через альтернативні пристрої.

5. Валідний HTML-код

Код у шаблонах Vue складається із валідної HTML-розмітки. Для перевірки її коректності було використано розширення HTML Validator для Google Chrome. У результаті тестування не було виявлено критичних помилок, що підтверджує технічну правильність структури сторінок. Це, своєю чергою, забезпечує кросбраузерну сумісність і коректне відображення інтерфейсу на різних пристроях.

6. Прогресивне покращення (progressive enhancement)

Інтерфейс побудований так, що навіть без JavaScript (або у випадку його часткової недоступності) основні елементи будуть відображатись коректно. Хоча функціонал без JS обмежений, базова структура, назви елементів і навігація залишаються доступними.

7. Масштабованість і підтримка майбутніх оновлень

Завдяки використанню Vue CLI та сучасної структури проєкту система легко масштабується. Кодова база організована в компоненти, що спрощує оновлення або впровадження додаткових функцій доступності в майбутньому (наприклад, ARIA Live Regions, автозаповнення, темна тема).

Завдяки поєднанню сучасного підходу до розробки, семантичної HTML-структури та базових атрибутів доступності, система демонструє відповідність принципу «Надійність». Це створює підґрунтя для сумісності з широким спектром платформ, браузерів і допоміжних технологій — як на поточному етапі, так і в процесі майбутнього розвитку.

3.3. Використання інструментів перевірки доступності (Lighthouse)

Щоб упевнитися, що система відповідає стандартам WCAG 2.1, у процесі розробки було використано інструменти автоматизованої перевірки доступності. Ці інструменти дозволяють виявити критичні бар'єри, попередити помилки на ранніх етапах та підвищити якість взаємодії з користувачем. Зокрема, були використані:

Lighthouse — це вбудований у Chrome DevTools інструмент, який надає оцінку за категоріями: Performance, Best Practices, SEO та Accessibility.

Результати сканування accessibility:

- Середній показник доступності: **100 балів**;
- Усі кнопки мали текст або aria-label;

- Усі інтерактивні елементи мали фокус-стили та були досяжні з клавіатури.

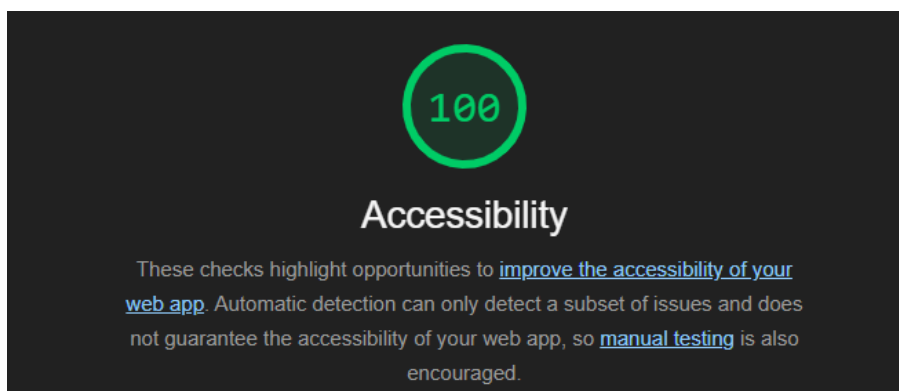


Рис. 3.2. Оцінка доступності Lighthouse

Джерело: розроблено автором

Крім автоматичних інструментів, були застосовані ручні перевірки, що включали:

- Клавіатурну навігацію: доступ до всіх кнопок, полів фільтрації та елементів списку книг через Tab / Shift+Tab;
- Імітацію скрінрідера: перевірка фокусів, семантики, назв елементів;
- Перевірку на мобільних пристроях — елементи коректно масштабуються, зберігають структуру та читабельність(див. Додаток А).

Застосування сучасних інструментів перевірки доступності — **Lighthouse** — дозволило комплексно оцінити рівень відповідності системи стандартам WCAG 2.1. Усі виявлені неточності були оперативно виправлені, що підтверджує високу якість реалізації доступності в межах розробленого вебінтерфейсу бібліотечної системи.

3.4. Юзабіліті-тестування із залученням користувачів

У рамках реалізації принципів доступності WCAG 2.1 у розробленій системі управління бібліотекою було проведено базове юзабіліті-тестування із залученням користувачів, з метою виявлення потенційних бар'єрів взаємодії з інтерфейсом та підтвердження зручності навігації.

Мета тестування

Основною метою було перевірити, наскільки ефективно реалізовані елементи доступності з погляду кінцевого користувача. Особливу увагу приділено навігації з клавіатури, зрозумілості структури сайту та помітності фокусу при перемиканні між елементами.

Учасники

У тестуванні взяли участь:

- один користувач без порушень зору, який протестував взаємодію з клавіатури (без миші);
- один користувач із досвідом використання скрінрідера (NVDA);
- один користувач, який загалом оцінював інтерфейс на предмет зручності й зрозумілості.

Усі учасники тестували систему у браузерях Google Chrome та Mozilla Firefox.

Сценарії тестування

Кожному користувачу було запропоновано виконати наступні дії:

1. Увійти до системи.
2. Перейти на головну сторінку.
3. Використати клавіші Alt+1, Alt+2, Alt+3 для перемикання між основними регіонами.
4. Виконати пошук книги.
5. Перейти до списку книг.
6. Прочитати назви та опис книг.
7. Вийти з облікового запису.

Результати

Користувач з клавіатурною навігацією:

- Усі комбінації клавіш (Alt+1, Alt+2, Alt+3) працювали коректно.
- Видимий фокус на елементах був чітким.
- Пошук можна було легко активувати з клавіатури.
- Загальна навігація — інтуїтивно зрозуміла.

Користувач зі скрінрідером NVDA:

- Заголовки були правильно ієрархізовані, що полегшило перемикання між секціями (H1, H2).
- Навігація читалася коректно завдяки aria-label.
- Кнопки без тексту (наприклад, “вийти”) мали відповідні aria-label, що зробило інтерфейс зрозумілим.
- Виявлено незначну затримку при перемиканні між компонентами — ймовірно, пов’язано з рендерингом Vue.

Загальний користувач:

- Позитивно оцінив шорткоди навігації.
- Ієрархія та логіка розташування елементів на сторінці — зрозумілі.
- Коментував, що інтерфейс “візуально чистий і передбачуваний”.

На основі отриманих відгуків можна зробити висновок, що реалізація принципів WCAG 2.1 була ефективною з точки зору реальних користувачів. Навігація з клавіатури, описові атрибути, правильна структура DOM та чіткий фокус забезпечили зручність взаємодії незалежно від способу навігації. Хоча тестування було обмежене за кількістю учасників, воно дозволило виявити та підтвердити успішність реалізованих рішень.

Висновки до розділу 3

У розділі 3 було детально розглянуто процес реалізації технічних рішень для забезпечення доступності веб-інтерфейсу системи управління бібліотекою відповідно до стандарту WCAG 2.1. Особливу увагу приділено організації клавіатурної навігації, візуальним індикаторам фокусу, а також коректному застосуванню семантичних елементів та атрибутів ARIA для покращення сприйняття інтерфейсу користувачами із різними потребами.

Проведений аналіз результатів автоматизованих аудитів доступності та юзабіліті-тестування засвідчив, що реалізовані рішення ефективно усувають основні бар'єри при використанні системи. Це створює передумови для комфортного користування інтерфейсом як звичайними користувачами, так і людьми з обмеженими можливостями.

Розроблена архітектура з компонентним підходом дозволяє легко масштабувати та підтримувати доступність на всіх рівнях системи, що є важливою перевагою для подальшого розвитку проєкту.

ВИСНОВКИ

У результаті виконання проєкту було розроблено веб-інтерфейс для системи управління бібліотекою, що відповідає ключовим вимогам стандарту **WCAG 2.1**. Основна мета — створення доступного інтерфейсу, яким можуть ефективно користуватися як звичайні користувачі, так і люди з обмеженими можливостями — була досягнута через імплементацію низки технічних рішень у межах чотирьох основних принципів доступності: **сприйнятність, керованість, зрозумілість, надійність**.

Зокрема:

- Використано **Vue.js 3** як основний фреймворк для розробки інтерфейсу, що дозволило реалізувати компонентний підхід та зручно підтримувати доступність на рівні окремих частин системи.
- Реалізовано **клавiатурну навігацію** через спеціальні комбiнації клавіш (Alt+1, Alt+2, Alt+3) для перемикання між основними регіонами сайту, що відповідає принципу *керованості*.
- Забезпечено чітке **фокусування елементів** у стилях, що дозволяє користувачу легко орієнтуватися в інтерфейсі без використання миші.
- Дотримано **ієрархії заголовків**, що є критичним для сприйняття структури сторінки користувачами скрінрідерів.
- Використано **атрибути ARIA**, такі як `aria-label`, `role` та інші, для уточнення ролі елементів і покращення озвучування.
- Проведено **юзабіліті-тестування**, яке підтвердило функціональність елементів доступності з боку реальних користувачів.

Крім того, для перевірки відповідності було застосовано **інструмент аудиту доступності**, такий як Lighthouse. Він засвідчив відсутність критичних бар'єрів для користувачів із порушеннями зору, рухової активності чи когнітивними особливостями.

Основні переваги реалізованого підходу:

- Простота підтримки: завдяки розділенню на компоненти та дотриманню стандартів, систему легко оновлювати та масштабувати.
- Гнучкість: реалізовані рішення можуть бути адаптовані для інших веб-проєктів.
- Інклюзивність: інтерфейс дружній до всіх категорій користувачів, що відповідає сучасним європейським та світовим вимогам щодо цифрової доступності.

Перспективи розвитку

Подальші напрямки розвитку системи включають:

- Додавання функцій персоналізації інтерфейсу (зміна розміру шрифтів, темна тема тощо).
- Реалізація підтримки мобільних пристроїв із забезпеченням мобільної доступності.
- Проведення більш масштабного юзабіліті-тестування з залученням користувачів із різними формами інвалідності.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

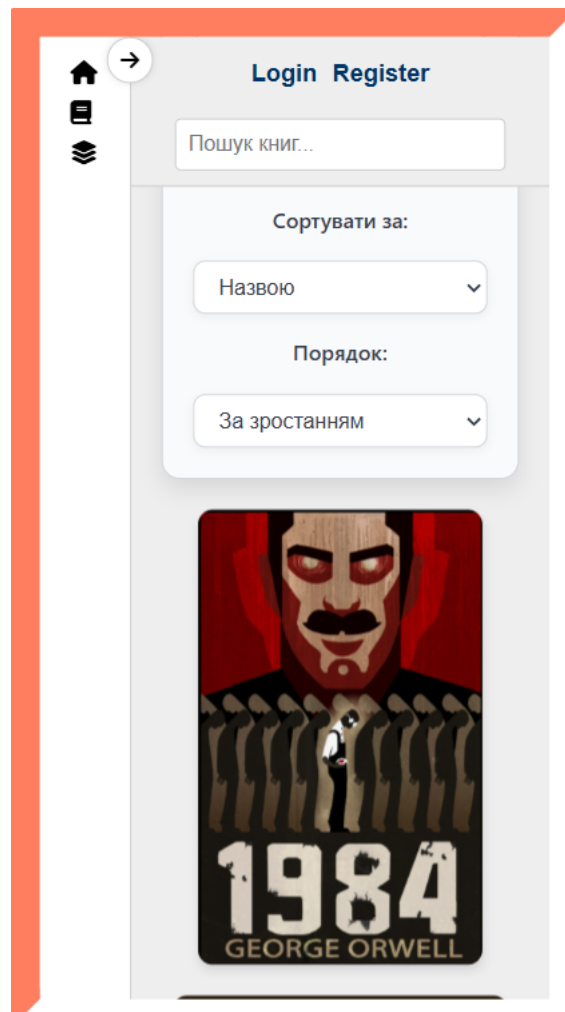
1. W3C. Web Content Accessibility Guidelines (WCAG) 2.1 [Електронний ресурс].
– Режим доступу: <https://www.w3.org/TR/WCAG21/> (дата звернення: 01.05.2025).
2. W3C. Introduction to Web Accessibility [Електронний ресурс]. – Режим доступу: <https://www.w3.org/WAI/fundamentals/accessibility-intro/> (дата звернення: 01.05.2025).
3. W3C. About WCAG [Електронний ресурс]. – Режим доступу: <https://www.w3.org/WAI/standards-guidelines/wcag/> (дата звернення: 01.05.2025).
4. W3C. Understanding Conformance Requirements: Level A [Електронний ресурс].
– Режим доступу: <https://www.w3.org/WAI/WCAG22/Understanding/conformance> (дата звернення: 01.05.2025).
5. W3C. How to Meet WCAG (Quick Reference) [Електронний ресурс]. – Режим доступу: <https://www.w3.org/WAI/WCAG22/quickref/> (дата звернення: 01.05.2025).
6. W3C. Understanding Conformance Requirements: Level AA [Електронний ресурс].
– Режим доступу: <https://www.w3.org/WAI/WCAG22/Understanding/conformance> (дата звернення: 01.05.2025).
7. W3C. Understanding Success Criterion 1.4.3: Contrast (Minimum) [Електронний ресурс].
– Режим доступу: <https://www.w3.org/WAI/WCAG21/Understanding/contrast-minimum> (дата звернення: 01.05.2025).
8. W3C. Understanding Conformance Requirements: Level AAA [Електронний ресурс].
– Режим доступу: <https://www.w3.org/WAI/WCAG22/Understanding/conformance> (дата звернення: 01.05.2025).

9. W3C. Quick Reference for AAA requirements [Електронний ресурс]. – Режим доступу: <https://www.w3.org/WAI/WCAG22/quickref/> (дата звернення: 01.05.2025).
10. United Nations. Convention on the Rights of Persons with Disabilities [Електронний ресурс]. – Режим доступу: <https://www.un.org/development/desa/disabilities/convention-on-the-rights-of-persons-with-disabilities.html> (дата звернення: 01.05.2025).
11. European Telecommunications Standards Institute (ETSI). EN 301 549 V3.2.1 (2021-03) [Електронний ресурс]. – Режим доступу: https://www.etsi.org/deliver/etsi_en/301500_301599/301549/03.02.01_60/en_301549v030201p.pdf (дата звернення: 01.05.2025).
12. Americans with Disabilities Act (ADA) [Електронний ресурс]. – Режим доступу: <https://www.ada.gov/> (дата звернення: 01.05.2025).
13. Australian Government. Web Content Accessibility Guidelines 2.0 (WCAG 2.0) Australia [Електронний ресурс]. – Режим доступу: <https://www.w3.org/TR/WCAG20/> (дата звернення: 01.05.2025).
14. Закон України "Про основи соціальної захищеності осіб з інвалідністю в Україні" [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/875-12#Text> (дата звернення: 01.05.2025).
15. Постанова Кабінету Міністрів України № 1088 від 21 листопада 2018 р. [Електронний ресурс]. – Режим доступу: https://zakon.rada.gov.ua/laws/main/149-2021-%D1%80?utm_source=chatgpt.com#Text (дата звернення: 01.05.2025).
16. ДСТУ ISO/IEC 40500:2015 (WCAG 2.0) [Електронний ресурс]. – Режим доступу: <https://www.kmu.gov.ua/storage/app/uploads/public/65d/329/d5d/65d329d5d584f046501885.pdf> (дата звернення: 01.05.2025).
17. Закон України "Про інформацію" [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/2657-12#Text> (дата звернення: 01.05.2025).

18. Закон України "Про електронні комунікації" від 16 грудня 2020 р. № 1089-IX [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/1089-20#Text> (дата звернення: 01.05.2025).

ДОДАТКИ

ДОДАТОК А



Додаток А. Вигляд веб-застосунку з мобільного пристрою