

АНАЛІЗ ДОСТУПНОСТІ ТА УДОСКОНАЛЕННЯ ВЕБ-ПРОДУКТУ ДЛЯ
КОРИСТУВАЧІВ З ОБМЕЖЕНИМИ МОЖЛИВОСТЯМИ НА ПРИКЛАДІ

[HTTP://WWW.NTU.EDU.UA/](http://www.ntu.edu.ua/)

ANALYSIS OF ACCESSIBILITY AND IMPROVEMENT OF A WEB PRODUCT FOR
USERS WITH LIMITED OPPORTUNITIES ON THE EXAMPLE OF

[HTTP://WWW.NTU.EDU.UA/](http://www.ntu.edu.ua/)



Безверхий Олександр Ігорович, доктор фізико-математичних наук, професор, Національний транспортний університет, професор кафедри інформаційних систем і технологій, o_bezver@ukr.net, тел. +380676849131, Scopus ID:6603638908, 14067133500

<https://orcid.org/0000-0002-0834-6335>



Донець Вероніка Василівна, старший викладач кафедри інформаційних систем і технологій, Національний транспортний університет, nerik.2008@ukr.net, тел. +38093686692

<https://orcid.org/0000-0003-2353-0699>

Анотація. Робота присвячена дослідженню доступності веб-продукту <http://www.ntu.edu.ua/> для користувачів з обмеженими можливостями з використанням допоміжних технологій. Проведено аналіз на всіх етапах життєвого циклу розробки програмного забезпечення на доступність згідно документу «Специфікація вимог до програмного забезпечення» та згідно до WCAG3[1,2,4,6]. Проведено аналіз тестування програмного забезпечення на доступність з використанням одного із інструментів автоматичного тестування доступності WAVE[9], паралельно з ним тестування клавіатури та програми зчитування з екрану NVDA. Проведено аналіз та надано рекомендації для усунення невідповідностей для веб-продукта <http://www.ntu.edu.ua/>, згідно до вимог WCAG21[7,8].

Ключові слова: доступність, дизайн, проектування, тестування, веб-сайт, допоміжні технології, адаптивні стратегії.

Вступ Майже кожна людина, ймовірно, в певний момент життя зазнає певну форму з обмежених можливостей – тимчасової чи постійної. Доступність означає, що всі люди можуть сприймати, розуміти, переміщатися та взаємодіяти з електронною інформацією, а також бути активними, брати участь у цифровому світі. Доступний дизайн покращує доступ до інформації в Інтернеті для людей з обмеженими можливостями та без них. Добре розроблені, доступні веб-сайти прискорюють доставку інформації та послуг. Вводити доступність потрібно вже на етапі аналізу програми, щоб вимоги які стосуються доступності були прописані в документі «Специфікація вимог до програмного забезпечення». Тоді на кожному етапі життєвого циклу розробки програмного забезпечення ми можемо контролювати та покращувати продукт на доступність для людей з обмеженими можливостями і цим покращувати зручність використання іншим користувачам.

Мета роботи: аналіз та удосконалення веб-сайту для людей з вадами зору, слуху, опорно-рухового апарату та когнітивними вадами на прикладі <http://www.ntu.edu.ua/>. Продукт має бути максимально доступним із застосуванням допоміжних технологій, таких як програм зчитування з екрану, контрастних режимів відображення контенту та застосування різних допоміжних методів для вводу тексту та переміщення по продукту.

Аналіз та удосконалення продукту

Ми провели аналіз веб-сайту <http://www.ntu.edu.ua/> на доступність для користувачів з обмеженими можливостями та до кожного типу невідповідності продукту до вимог WCA21 були залишені пропозиції, рішення, приклади реалізації для удосконалення продукту. Через відсутність доступу до коду вебпродукту ми наведемо приклади імплементованих рішень з amazon.com, у якого за результатами аудиту сайт відповідає на 95% до вимог доступності за WCAG21. Для тестування ми вибрали широковживану операційну систему Windows 10 та інтернет-браузер Chrome версії 101.0.4951.67.

1. Головна сторінка. 1.1. Доступність на сприйняття контенту

Почнемо наш аналіз з запуском автоматизованого онлайн інструменту оцінки веб-доступності, WAVE або AXE, та для тестування семантичної структури, рівнів заголовків, списків та елементів форм, пов'язаних із групою. Валідатор видав нам наступний «Підсумок»: • 55 помилок; • 14 помилок в рівні контрастів; • 35 попереджень • Та 40 структурних елементів, які ми також детально розглянемо. Якщо починати зверху сторінки, то перша невідповідність належить до вимоги рівня А, «3.1.1 Мова сторінки». Тобто, мова документа не визначена, не проставлене значення атрибуту, як на рисунку 1.

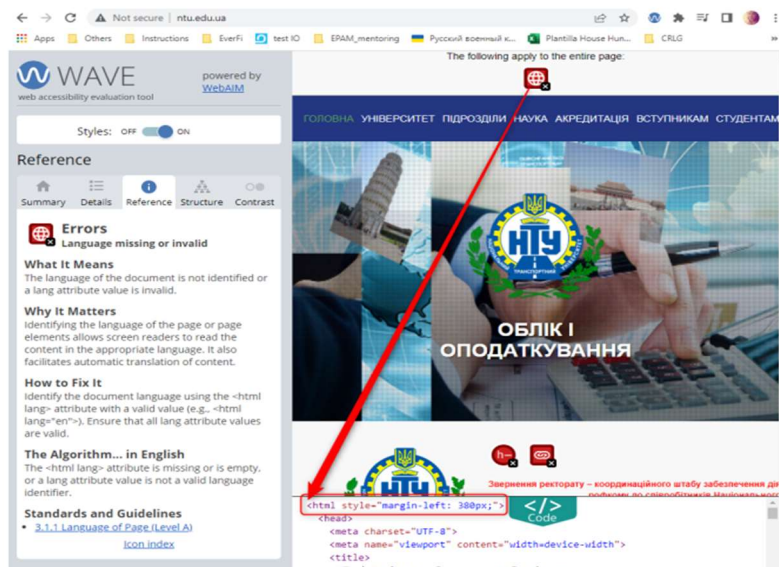


Рисунок 1 – Мова документа не визначена
Figure 1 – The language of the document is not defined

Визначення мови сторінки або елементів сторінки дозволяє програмам зчитування з екрану читати вміст відповідною мовою. Це також полегшує автоматичний переклад вмісту.

Рішення: Визначити мову документа за допомогою атрибута з дійсним значенням (наприклад), після імплементатії валідатор вже ідентифікував що це не помилка, як це зображено на рисунку 2.

Інша проблема це пустий елемент, зображений на рисунку 3., якому присвоєно значення заголовку першого рівня тегом, що порушує наступні вимоги до WCAG21: • 1.3.1 Інформація та відносини (рівень А), • 2.4.1 Блоки обходу (рівень А), • 2.4.6 Заголовки та мітки (рівень AA), та посилання, яке не має контексту, і порушує іншу вимогу WCAG21 рівня А, «2.4.4 Призначення посилання (в контексті)».

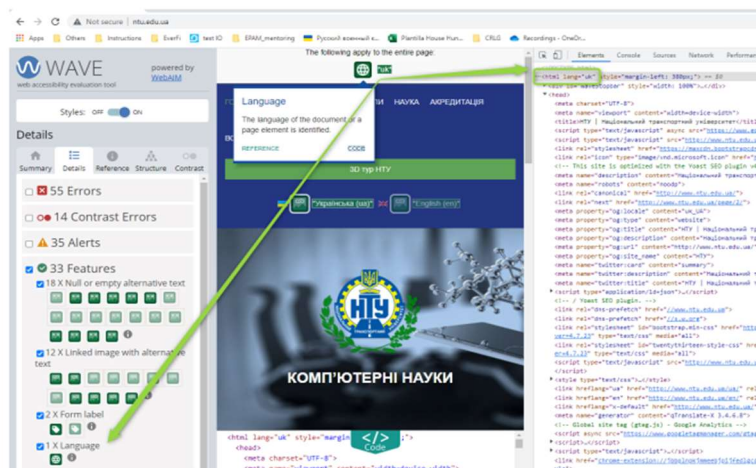


Рисунок 2 – Мова документа визначена.
Figure 2 – The language of the document is defined

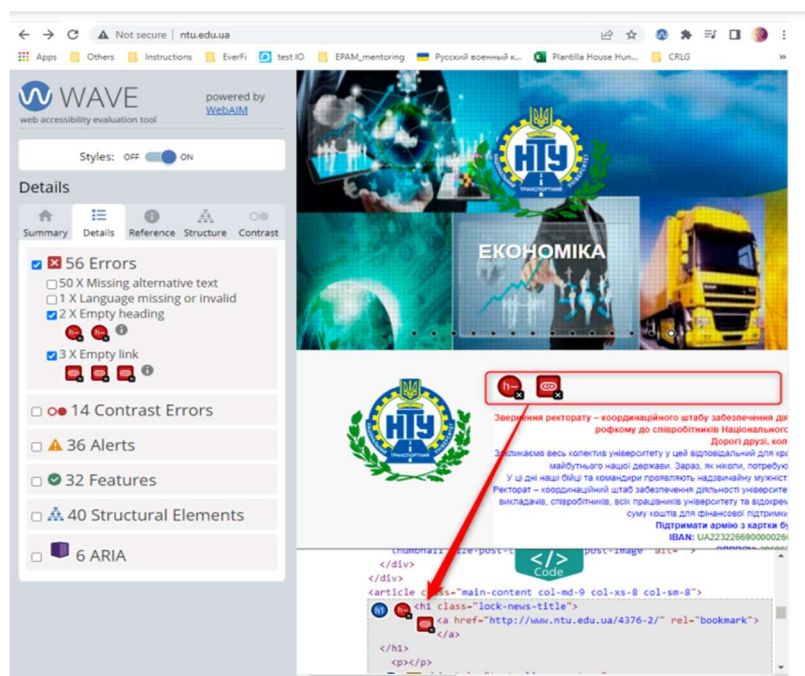


Рисунок 3 – Пустий заголовок та пуста лінка
Figure 3 – An empty header and an empty link

Розглянемо перший випадок, де порожнє посилання. Це значить, що посилання не містить тексту. Якщо посилання не містить тексту, функція або призначення посилання не буде представлено користувачеві. Це може спричинити плутанину для користувачів клавіатури та програми зчитування з екрана.

Рішення: видалити порожнє посилання або ввести текст у посиланні, який описує функції та/або ціль цього посилання. Використовуючи наступний алгоритм: елемент прив'язки має атрибут `<href>`, але не містить тексту (або лише пробіли) і не містить зображень з альтернативним текстом.

Розглянемо другий випадок, де заголовок не містить вмісту. Деякі користувачі, особливо користувачі клавіатури та програми зчитування з екрана, часто переміщаються за елементами заголовку. Порожній заголовок не надасть жодної інформації та може внести плутанину.

Рішення: потрібно переконаватися, що всі заголовки містять інформативний вміст, а якщо ні, то потрібно видалити тег `<h1>`. Але зауважимо, що заголовку краще не містити посилання, бо це може

заплутати користувача з обмеженими можливостями, якщо він буде переглядати сторінку по заголовках використовуючи клавішу з клавіатури «Н» при включеній програмі зчитування, він подумає, що це просто як текст, який не містить посилання. Схожа ситуація з заголовком спостерігається і для елемента `/html/body/div[3]/div/div/article/h6[2]` (елемент на сторінці за XPath).

1.2. Рівень контрастів

Як згадувалось раніше, «Підсумок» WAVE нам повідомив про 14 випадків дуже низького контрасту між кольорами тексту та фону, що порушує вимогу рівня А до WCAG21: «1.4.3 Контраст (мінімум) (рівень AA)». Ці невідповідності присутні в наступних елементах: • посилання «Головна»; • посилання «3D тур НТУ»; • посилання, котрим задано колір попереднього плану `#FF0000`; • посилання, котрим задано колір попереднього плану `#3366FF`; • посилання, котрим задано колір попереднього плану `#00A0F0`; • заголовок рівня «МІЖНАРОДНЕ СПІВРОБІТНИЦТВО». Адекватна контрастність тексту необхідна всім користувачам, особливо користувачам із слабким зором. **Рішенням** до цієї невідповідності є збільшення контрасту між кольором переднього плану (тексту) і кольором фону. Великий текст (більше 18 пунктів або 14 пунктів напівжирний) не вимагає такого контрасту, як менший текст. Присутній текст із коефіцієнтом контрастності менше 4,5:1 або великий текст (більше 18 пунктів або 14 пунктів жирним шрифтом) має коефіцієнт контрастності менше 3:1. WCAG вимагає, щоб елементи сторінки мали визначені (або успадковані) кольори як переднього плану, так і фону, які забезпечують достатній контраст. Коли текст представлено на фоновому зображенні, текст повинен мати визначений колір фону (зазвичай у CSS), який забезпечує достатній контраст тексту, коли фонове зображення вимкнено або недоступне. WAVE не визначає проблеми контрастності в тексті з прозорістю CSS, градієнтами або фільтрами.

1.3 Структура сторінки

Одною із важливих проблем, з якою зіштовхується наш користувач, це невідповідність структури заголовків, поєднання заголовка з посиланням. Проаналізувавши структуру заголовків для нашого продукту, ми виявили низку невідповідностей до вимоги WCAG21 рівня А, «1.3.1 Інформація та відносини», частину відображено на рисунках 4.4 та 4.5: • заімплементовано більше одного заголовка з тегом `<h1>`; • заголовки присвоєні не в логічному порядку.

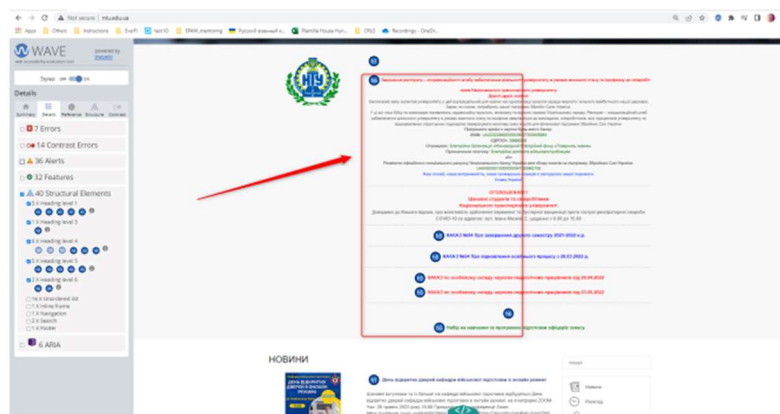


Рисунок 4 – Заголовки присвоєні не в логічному порядку

Figure 4 – Headings are not assigned in a logical order

Отже, найважливіший заголовок має перший ранг (`<h1>`) і він має бути єдиним на веб-сторінці. Заголовки з рівним або вищим рангом починають новий розділ, заголовки з нижчим рангом починають нові підрозділи, які є частиною розділу вищого рангу. Однією з головних причин того, що елемент `<h1>` має з'являтися на початку основного вмісту, є те, що користувачі програми зчитування з екрана можуть використовувати комбінації клавіш для переходу безпосередньо до першого елемента `<h1>`, що, в принципі, повинно дозволити їм перейти безпосередньо до основного елемента вмісту веб-сторінки. Якщо елемента `<h1>` немає або якщо елемент `<h1>` з'являється десь не на початку основного вмісту, користувачам програми зчитування з екрана доведеться прослухати додатковий вміст веб-сторінки, щоб зрозуміти структуру сторінки, витрачаючи дорогоцінний час. Пропуск рядків заголовків

може ввести в оману і його слід уникати, якщо це можливо: потрібно переконатися, що за <h2> не слід використовувати безпосередньо <h4>, наприклад. Можна пропускати ранги під час закриття підрозділів, наприклад, <h2>, що починає новий розділ, може слідувати за <h4>, оскільки він закриває попередній розділ. Розмітьте підрозділи сторінки як елементи заголовка рівня <h2>. Якщо в розділах рівня 2 є підрозділи, позначте ці розділи як елементи заголовка рівня <h3>, тощо. Все, що стоїть перед основним вмістом сторінки, взагалі не повинно бути позначено жодними заголовками, хоча це не залізне правило. Наведемо приклади імплементованих рішень з amazon.com, у якого за результатами аудиту сайт відповідає на 95% до вимог доступності за WCAG21. То ж, зверніть увагу на приклад розташування заголовків з сайту Amazon на рисунку 6.

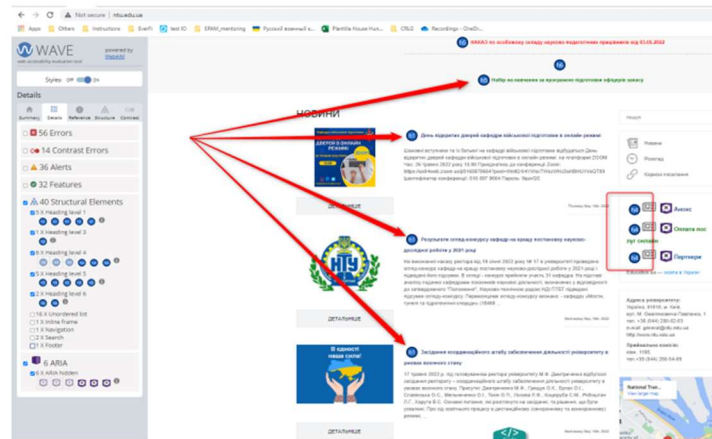


Рисунок 5 – Більше одного заголовка з тегом <h1>
Figure 5 – More than one heading with <h1> tag

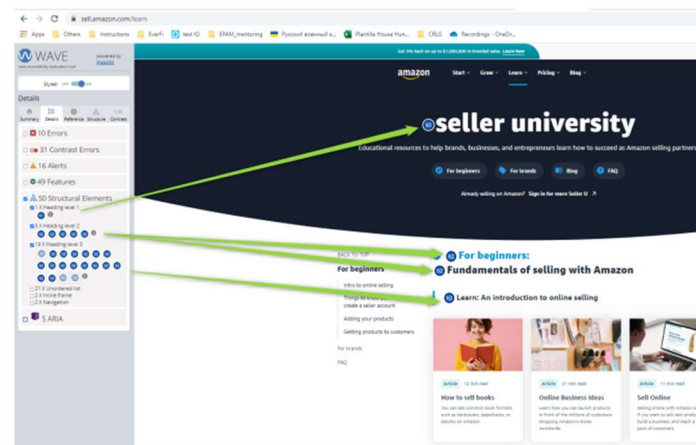


Рисунок 6 – Коректна структура заголовків
Figure 6 – Correct header structure

1.4 Навігація по інтерактивних елементах з клавіатури.

Також є вимоги для доступності клавіатури. Користувачі клавіатури повинні мати доступ до всіх елементів інтерфейсу користувача та працювати з ними (миша не потрібна). Веб-сторінка має забезпечувати чіткий, послідовний видимий індикатор фокусу, щоб користувачі клавіатури завжди могли визначити, який елемент має фокус. Отже, результат тестування показав, що користувач з обмеженими можливостями має доступ до всіх інтерактивних елементів, але ми виявили тільки одну помилку, це погано видимий або відсутній контур фокуса навігації як на рисунку 7.

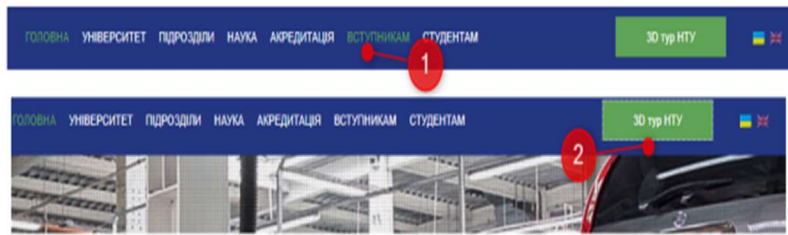


Рисунок 7 – Відсутній та погано видимий контур фокуса навігації.

Figure 7 – Missing and poorly visible navigation focus outline.

Рішенням може бути нижче наведений лістинг це CSS, який змінює колір фону для вищевказаних елементів, коли вони отримують фокус миші або клавіатури. #mainnav a:hover, #mainnav a:active, #mainnav a:focus { background-color: #DCFFFF; color:#000066; }

1.5 Зчитування з екрану

Для тестування покриття вимог до WCAG21 рівня А, «4.1.2 Назва, Роль, Значення: Для всіх компонентів інтерфейсу користувача (включаючи, але не обмежуючись: елементи форми, посилання та компоненти, створені за допомогою скриптів), ім'я та роль можуть бути визначені програмно; стани, властивості та значення, які можуть бути встановлені користувачем, можуть бути встановлені програмно; і сповіщення про зміни до цих елементів доступне для агентів користувача, включаючи допоміжні технології» ми використовуємо безкоштовний і один із найпопулярніших програм зчитування з екрану – «NVDA screen reader». Як результат тестування, функціонал на веб-сторінці покритий даною вимогою максимально. Всі інтерактивні елементи, ролі та значення озвучуються згідно до вимоги WCAG21:4.1.2, як результат можна переглянути лог NVDA на рисунку 8.

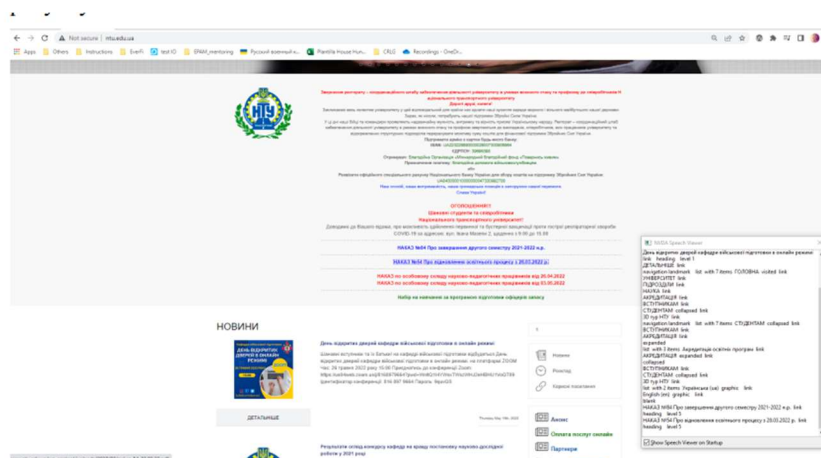


Рисунок 8 – Зчитування інтерактивних елементів

Figure 8 – Reading interactive elements

Також всі неінтерактивні елементи: заголовки, тексти та інший контент також озвучується згідно до вимоги WCAG21:4.1.2, як з використанням клавіш вниз/вверх, так і з використанням базових комбінацій клавіш, результат можна переглянути в логі NVDA на рисунку 9.

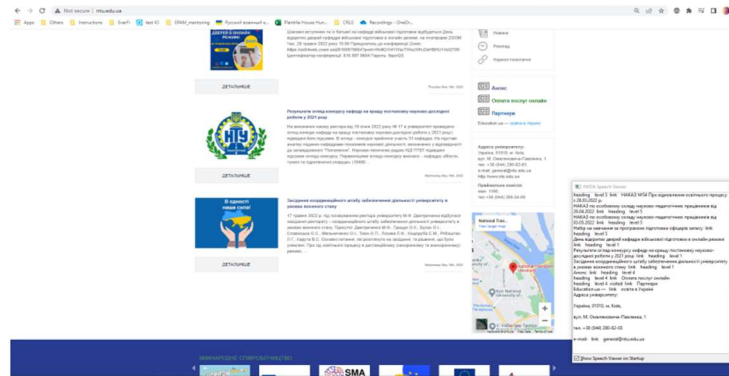


Рисунок 9 — Зчитування контенту в режимі браузера
Figure 9 — Reading content in browser mode

2. Корисна інфографіка. Медіа файли

Один із важливих та часто вживаних неінтерактивних елементів на вебсайті є картинки, заімплементовані в html з тегом `<img>`. Одна із веб-сторінок на нашому продукті повністю заповнена картинками з текстом, які несуть змістовий та важливий контент для користувачів. Для перевірки картинки на доступність ми використали автоматизований інструмент оцінки веб-доступності WAVE та для перехресної перевірки зчитування з екрану – NVDA. Автоматизований валідатор відобразив результат, що для елемента присвоєно атрибут «alt=""», тобто null, це не було б помилкою, якщо зображення несло б декоративний сенс. Але так як вся веб-сторінка присвячена зображенням з текстом, то це критична помилка, що атрибут «Alt» пустий, результат перевірки зображено на рисунку 10.

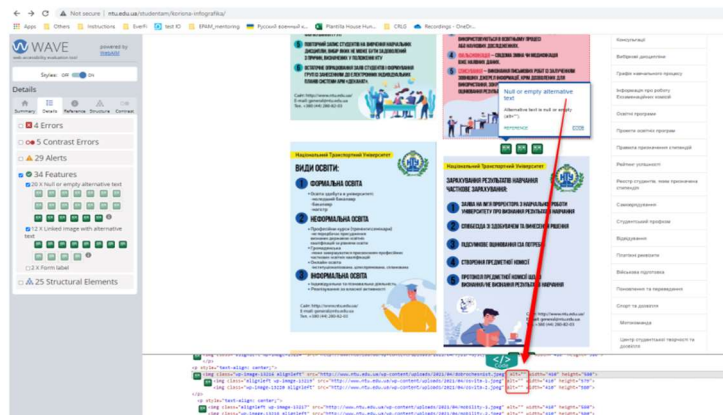


Рисунок 10 – Результат автоматизованої перевірки на доступність елемента «Alt» є скороченням «альтернативного тексту».

Figure 10 — The result of an automated check for the availability of the element "Alt" is short for "alternative text".

Це короткий письмовий опис зображення, який має сенс у зображенні, якщо його з якихось причин неможливо переглянути. Добре написаний альтернативний текст важливий для доступності вашого веб-сайту та його пошукової оптимізації. Результат того, як інструмент зчитування з екрану NVDA в режимі браузера, при використанні стрілок вгору/вниз, буде озвучувати нашому користувачу з обмеженими можливостями контент зображення, відображено на рисунку 11.



Рисунок 11 – Результат зчитування елемента за допомогою NVDA

Figure 11 – The result of reading the element using NVDA

То ж було прийнято рішення додати текст до атрибуту «Alt». Результат вище запропонованих змін перевірили за допомогою програми зчитування з екрану NVDA, і як результат, наш користувач з обмеженими можливостями вже чує що на веб-сторінці відображено зображення, які передають певний зміст, див. рисунок 12.



Рисунок 12 – Результат зчитування елемента за допомогою NVDA після додавання тексту до атрибуту «alt».

Figure 12 – The result of reading the element using NVDA after adding text to the "alt" attribute.

Було запропоновано декілька корисних пропозицій, це дати доступ до всієї інформації, яка відображена на зображеннях «Корисної інфографіки»: • додати текстовий весь контент із зображення під ним, в вигляді тексту; • додати весь контент із зображення до атрибуту «Alt»; • додати посилання з назвою «Більше інформації про *назва зображення*» на веб-сторінку, де в вигляді тексту прописана інформація із зображень.

Даними пропозиціями ми покрили вимогу рівня А до WCAG21, 1.1.1 «Нетекстовий зміст: Весь нетекстовий зміст, який надається користувачеві, має текстову альтернативу, яка служить еквівалентній меті, за винятком ситуацій, наведених нижче». Незважаючи на те, що це одна з найбільших проблем, які впливають на доступність Інтернету, ми продовжуємо бачити різні та неправильні методи впровадження альтернативного тексту в Інтернеті. Якби спільнота розробників могла повністю прийняти еквівалентний альтернативний текст, Інтернет був би набагато доступнішим місцем!

Висновки

1. На всіх етапах життєвого циклу розробки програмного забезпечення потрібно працювати над доступністю. Документ «Специфікація вимог до програмного забезпечення» потрібно відразу оформлювати згідно до WCAG3 з самого першого етапу аналізу програмного забезпечення. 2. Для тестування програмного забезпечення на доступність потрібно використовувати один із інструментів автоматичного тестування доступності WAVE, паралельно з ним клавіатуру та програму зчитування з

екрану NVDA. 3. Проведено аналіз, на доступність веб-продукту <http://www.ntu.edu.ua/> для користувачів з обмеженими можливостями з використанням допоміжних технологій. 4. Надано рекомендації для усунення невідповідностей для веб-продукту <http://www.ntu.edu.ua/>, згідно до вимог WCAG21.

Перелік посилань

1. <https://visuresolutions.com/uk/software-requirement-specification-srs-tips-template/>
2. Вигерс К.И. Разработка требований до ПЗ. – М.: Російська редакція Microsoft, 2004. – 575 с.
3. Guckenheimer S., Peter J. Software Engineering With Microsoft Visual Studio. Team System. – Addison Wesley, 2006. – 273 p.
4. http://baklaniv.at.ua/ANALIZ_VYMOG/lekciya_1-2.pdf
5. <https://artsandculture.google.com/?hl=uk>
6. WCAG Compliance - Web Accessibility Tester - siteimprove.com
7. Digital Guide to WCAG & EAA - Free Web Accessibility Guide - <https://www.crownpeak.com/>
8. Web Content Accessibility Guidelines (WCAG) 2 Level AA ...<https://www.w3.org> > WAI
9. Проверка доступности сайта: обзор инструментов для accessibility testing - <https://dou.ua/>

ANALYSIS OF ACCESSIBILITY AND IMPROVEMENT OF A WEB PRODUCT FOR USERS WITH LIMITED OPPORTUNITIES ON THE EXAMPLE OF HTTP://WWW.NTU.EDU.UA/

Bezverkhy Oleksandr I., Doctor of Physical and Mathematical Sciences, Professor, National Transport University, Professor of the Department of Information Systems and Technologies, o_bezver@ukr.net, tel. +380676849131, <https://orcid.org/0000-0002-0834-6335>, Scopus ID: 6603638908, 14067133500

Donets Veronika V., National Transport University, senior lecturer of the Department of Information Systems and Technologies, nerik2008@ukr.net, tel. +38093686692, <https://orcid.org/0000-0003-2353-0699>

Abstract. The work is devoted to researching the accessibility of the web product <http://www.ntu.edu.ua/> for users with disabilities using assistive technologies. An analysis was carried out at all stages of the software development life cycle for accessibility according to the document "Specification of requirements for software" and according to WCAG3[1,2,4,6]. An analysis of software accessibility testing was conducted using one of the WAVE[9] automatic accessibility testing tools, along with keyboard and NVDA screen reader testing. The analysis was carried out and recommendations were made to eliminate inconsistencies for the web product <http://www.ntu.edu.ua/>, in accordance with the requirements of WCAG21[7,8].

Key words: accessibility, design, engineering, testing, website, assistive technology, adaptive strategies.

Referens

1. <https://visuresolutions.com/uk/software-requirement-specification-srs-tips-template/>
2. Vigers K.I. Development of software requirements.- М.:Russian edition of Microsoft, 2004. - 575p. [in Ukrainian].
3. Guckenheimer S., Peter J. Software Engineering With Microsoft Visual Studio. Team System. – Addison Wesley, 2006. – 273 p. [in English].
4. http://baklaniv.at.ua/ANALIZ_VYMOG/lekciya_1-2.pdf
5. <https://artsandculture.google.com/?hl=uk>
6. WCAG Compliance - Web Accessibility Tester - siteimprove.com
7. Digital Guide to WCAG & EAA - Free Web Accessibility Guide - <https://www.crownpeak.com/>
8. Web Content Accessibility Guidelines (WCAG) 2 Level AA ...<https://www.w3.org> > WAI
9. Website accessibility check: overview of accessibility testing tools - <https://dou.ua/>. [in Russian].