

ОСОБЛИВОСТІ РОЗРОБКИ ТА АДАПТАЦІЇ WEB-ДОДАТКІВ ДЛЯ ЛЮДЕЙ З ОБМЕЖЕНИМИ МОЖЛИВОСТЯМИ

Вінницький Національний Технічний Університет

Анотація.

Дана робота присвячена огляду методів адаптації Web-додатку для користувачів з вадами зору та опорно-руховим апаратом. В даній роботі виконано дослідження і аналіз сучасного стану "accessibility" в галузі web-розробки. В роботі розглянуто етапи розробки сайтів та web-додатків та їх вплив і роль в підвищенні доступності для користувачів. Після сортування певного діапазону технологій було обрано оптимальний стек за певними параметрами. Розглянуто способи та стандарти для підвищення доступності сайтів.

Ключові слова: Web-додаток, доступність.

Abstract.

This paper is devoted to a review of methods of adapting Web-application for visually impaired users and musculoskeletal system. impact and role in increasing accessibility for users. After sorting a certain range of technologies, the optimal stack was selected according to certain parameters. Methods and standards for increasing the accessibility of sites are considered.

Keywords: Web-application, accessibility.

Вступ

У момент переходу на віддалений режим роботи більшість власників бізнесу розуміє, що настав час створювати цифровий продукт, наприклад, освітній сайт або інтернет-магазин. Один з варіантів розробки такого товару - створення веб-додатку. Це повноцінна програма, яку клієнти використовують через браузер. Обрана тема роботи є актуальною, оскільки попит на розробку Інтернет сервісів зростає, а значною частиною цих сервісів є саме web-додатки. Відповідно до зросту кількості додатків росте і кількість користувачів у таких сервісах у числі яких люди з обмеженими можливостями. Це говорить про доцільність та необхідність розробки додатків з достатньо функціональними та простими інтерфейсами та адаптованими під потреби усіх людей, якими могла б користуватись будь-яка людина.

Сучасний стан доступності у web-розробці

Для веб-сайтів органів державної влади, органів місцевого самоврядування та територіальних громад, а особливо для веб-сайтів через які можна замовити отримання публічних та адміністративних послуг важливим є питання доступності для усіх людей. Питання стандартів або методичних рекомендацій щодо розробки адаптованих веб-сайтів в Україні залишається не вирішеним. Переважно мій досвід упровадження доступності стосується закордонних проектів. Проте протягом останнього року я мав можливість працювати над проектом українською, що спонукало мене дізнатися більше про стан доступності в Україні.

На жаль, у нас не все так просто. Незважаючи на те, що 6,1 населення має інвалідність (2 800 600 осіб), 224 000 робочих днів на рік люди непрацездатні через виробничі травми, кількість побутових травм не визначена (не думаю, що вона нижча, ніж в інших країнах), а в країні триває війна й ми щодня дізнаємося про нові травми. Україна — порівняно малий ринок, і компаніям не цікаво вкладати в українську. Зокрема ні VoiceOver, ні Windows не мають підтримки українською. Здебільшого скрінрідери українською стають доступними завдяки ентузіастам й open source, зокрема до NVDA розробили синтезатор українською. Радує те, що є якісний синтез української на Android (Google TalkBack). Веб-доступність критична для українців з вадами зору, оскільки

дисплеї Брайля дорогі навіть в США, а для середньостатистичного українця сума \$3500 — дуже велика.

На сьогодні можна виділити два основних документи, визнаних на міжнародному рівні, що визначають стандарти та рекомендації доступності веб-сайтів для людей з обмеженими можливостями, це:

– The Web Content Accessibility Guidelines 2.0 (WCAG 2.0) – остання версія методичних рекомендацій розроблена міжнародним консорціумом World Wide Web Consortium (W3C). W3C – це міжнародна організація, що розробляє відкриті стандарти для мережі Інтернет, створює специфікації для HTML, CSS, тощо. Консорціум W3C створив окрему ініціативу щодо доступності веб-ресурсів «Web Accessibility Initiative» (WAI), метою якої є саме розробка стандартів доступності для веб-браузерів, засобів розробки веб-контенту, тощо для людей з обмеженими можливостями. Результатом роботи робочої групи по вебконтенту ініціативи WAI стала розробка зазначених методичних рекомендацій щодо доступності веб-сайтів – WCAG.

– Розділ 508 Закону про Реабілітацію США – зазначений закон з'явився у 1973 році та був першою законодавчою ініціативою створення рівних умов для людей з обмеженими можливостями. У 1998 році у розділ 508 зазначеного закону були внесені зміни, що заборонили федеральному уряду закупати товари та послуги, що не є повністю доступні для людей з обмеженими можливостями, включаючи послуги веб-дизайну. На базі цього розділу законодавчого акту були створені відповідні стандарти, опубліковані 21 грудня 2000 року.

Основний принцип закладений у зазначених вище зазначених методичних рекомендаціях та стандартах документів – це максимальна адаптація веб-сайтів до інструментів доступності, таких як текстове озвучування або використання без прив'язки до кольору [1]. Це включає в себе:

- Заповнення атрибутів альтернативного тексту (alt) для всіх графічних елементів, що мають не декоративну функцію, з чітким описом зображення та його функції.
- Додання окремих файлів субтитрів для всіх аудіо та відео-матеріалів.
- Спрощення та коректне створення таблиць, з чітким зазначенням рядків та стовпчиків заголовків та відповідності комірок таблиці заголовкам.
- Адаптація для перегляду сторінки без прив'язки до кольорів.
- Створення форм внесення інформації з послідовними логічними переходами між полями, наявністю всіх підписів та підказок щодо заповнення.

Етапи розробки web-додатку та їх вплив на «доступність» кінцевого продукту

1) Розробка технічного завдання. Цей етап включає в себе складання документації, що детально описує структурну та поведінкову моделі проекту [2]. Цей етап мало як впливає на доступність додатку, але при виконанні цього етапу можна спростити ідею певного функціоналу або його модернізацію, що може підвищити доступність продукту для користувачів. Простою мовою: описуємо коротко, технічною мовою, як виглядатиме і як працюватиме сайт або програма.

2) Створення дизайну та навігації. Цей етап включає в себе детальне промальовування та визначення позиціонування, типових структурних елементів проекту. Наприклад: створення ескізу сторінки сайту, розробка логотипу, вибір місць розташування елементів навігації, оформлення шрифтів та заголовків та багато іншого. На цьому етапі можна продумати хорошу та якісну кольорову гаму, вибрати оптимальні розміри елементів та шрифтів. Такі фактори напряму впливають на доступність додатку

3) Верстка – це опис типових структурних елементів проекту мовою розмітки, відповідно до заданої специфікації. Простіше кажучи: те, що намалював дизайнер, верстальник приводить до робочого вигляду, використовуючи мови розмітки HTML, XML та інші [2]. Цей етап найбільше впливає на рівень доступності проекту, тому що саме при верстці у додатку формується його структура, його наповнюють тегами, а теги наповнюють усіма відповідними атрибутами. У наш час це основні методи підвищення доступності у web-розробці.

4) Програмування - опис структурної та поведінкової моделей проекту мовами програмування.

Програміст дарує життя веб додаткам та сайтам, вибираючи найкращі алгоритми для вирішення задачі [2]. Під час цього етапу можна доповнити додаток деякими інтерактивними методами підвищення доступності. Наприклад, фокусування на важливому полі вводу для інформації або автоматичне відкриття меню при потребі.

5) Встановлення. Копіювання проекту на робочий сервер та інсталяція. Налаштування конфігурації сервера для працездатності сайтів та веб-додатків. Під час даного етапу не можна ніяк вплинути на рівень доступності додатку.

6) Тестування та налагодження. Виявлення синтаксичних та логічних помилок проекту та подальше їх усунення. Тестувальник знаходить несправності, складає список і відправляє його на доопрацювання розробникам. Ті, своєю чергою, виправляють помилки. Цей етап є заключним і може допомогти виправити помилки, які були допущені при роботі на попередніх етапах, включаючи і недопрацювання у плані доступності додатка.

Стандарт WAI-ARIA

WAI-ARIA — стандарт доступності активних Інтернет-додатків, визначає підходи до того, щоб зробити вміст сайтів та інтернет-програми доступнішими для людей з обмеженими можливостями.

Застосування стандарту особливо ефективно для сайтів з динамічним вмістом і розвиненими елементами управління інтерфейсом, розробленими з використанням Ajax, HTML, JavaScript, і пов'язаних з ними технологій. В даний час деякі функціональні можливості веб-сайтів недоступні для частини користувачів з обмеженими фізичними можливостями. WAI-ARIA задає нові шляхи надання функціональних можливостей програм-помічників. За допомогою WAI-ARIA розробники можуть створювати прогресивні веб-програми, доступні та зручні для всіх користувачів.

Стандарт WAI-ARIA задає набір атрибутів HTML, що дозволяють ідентифікувати функції, що взаємодіють з користувачем, встановити їх взаємозв'язок і поточне розташування. Стандарт описує нові техніки навігації для виділення областей та основних елементів структури сайту, таких як меню, основний вміст, другорядний вміст, інформація про банери. Наприклад, за допомогою WAI-ARIA розробники можуть позначити області на сторінках, дозволяючи користувачам легко переміщатися між областями за допомогою клавіатури без необхідності натискати клавішу Tab.

WAI-ARIA включає технології для позначення елементів управління, активних областей AJAX, подій і саморобних елементів управління. Техніки WAI-ARIA застосовні до кнопок, що випадають списків, календарів, деревоподібних елементів та ін .

Стандарт WAI-ARIA надає авторам сайтів:

— Ролі для опису типу активного елемента, наприклад: "menu," "treeitem", "slider," і "progressmeter". Наприклад, `<div role = "checkbox">`

— Ролі для опису структури сторінок, наприклад: заголовки, області та таблиці.

— Атрибути для опису стану, в якому знаходиться елемент керування, наприклад checked для чекбоксу і haspopup для меню.

— Атрибути для завдання активних областей сторінки, що дозволяють оновлення (наприклад, біржові котирування), а також спосіб появи даних оновлень. Наприклад, важливі оновлення можуть бути представлені в діалоговому вікні, а другорядні - з'являтися прямо на сторінці.

— Атрибути для перетягування об'єктів, що описують ресурси, що переносяться, і цільові положення в документі. Можливість керування з клавіатури об'єктами та подіями.

Основні методи та рекомендації для перевірки додатку на доступність відповідно міжнародним рекомендаціям.

Перевірити веб-сайт на відповідність зазначеним вище міжнародним методичним рекомендаціям та на доступність для людей з вадами зору можна наступним чином:

— Перевірити на відповідність спрощеним контрольним спискам у відповідності до зазначених вище стандартів та методичних рекомендацій: контрольний список WCAG 2.0 (англ.), контрольний список Розділу 508 Закону про Реабілітацію. Слід зазначити, що контрольні списки мають рекомендаційний характер та не є офіційними стандартами.

— Перевірити з використанням читачів тексту, таких як JAWS Screen Reader, NonVisual Desktop Access (NVDA), VoiceOver.

Рекомендації по перевірці адаптації веб-сайтів для людей з вадами зору Наведені нижче рекомендації по адаптації веб-сайтів для людей з вадами зору базуються на чеклисті статті Creating an ADA-compliant website (Nicole Nash).

1. Атрибут «alt». Кожне зображення, відео-файл, аудіо-файл, плагін, тощо, мають опис у атрибуті «alt». Складні графічні елементи мають окремий текстовий опис. Опис у атрибуті «alt»

містить призначення об'єктів. Якщо зображення використовується як посилання, переконайтесь, що у атрибуті «alt» є як опис зображення так і відомості про адресу переходу за посиланням. Декоративні графічні елементи, що не несуть ніяких функцій мають порожній атрибут «alt» (alt="")

2. Аудіо та відео-елементи Всі відео-елементи мають підписи. Всі аудіо-елементи мають опис. Створіть текстові субтитри до аудіо та відео-файлів. Замість влаштування відео з іншого веб-сайту у веб-сторінку робіть посилання на відео. Додайте посилання для завантаження медіа-програвача. Додайте окреме посилання на файл з субтитрами

3. Карти зображень. Сторінка повинне мати альтернативні посилання для карти зображень. Теги мають містити атрибут «alt»

4. Таблиці. Таблиці з даними мають рядки та/або стовпчики з заголовками що відповідно ідентифіковані з використанням тегу. Таблиці що використовуються лише для розміщення елементів НЕ ПОВИННІ мати рядків/стовпчиків з заголовками (тег). Комірки таблиці асоційовані з відповідними заголовками таблиці (наприклад за допомогою HTML атрибутів id, headers, scope та/або axis)

5. Блимаючі зображення. Переконайтесь що сторінка не містить повторюваних блимаючих зображень. Переконайтесь що сторінка не містить елементів з ефектом стробоскопа

6. Плагіни та скрипти. Якщо потрібне завантаження плагіну – сторінка повинна містити посилання на адаптовану для людей з вадами зору сторінку де його можна завантажити. Переконайтесь що всі Java-аплети, скрипти та плагіни і їх зміст (включаючи файли PDF та презентації PowerPoint, тощо) – доступні з використанням допоміжних технологій для людей з вадами зору або іншими альтернативними засобами доступу до такого контенту

7. Форми. Текстові поля для вводу інформації в формах повинні мати заповнений елемент «label». Якщо текст не доступний використовуйте атрибут «title». Усі спеціальні вказівки по заповненню полів мають бути внесені у назви полів. Переконайтесь що поля форм мають послідовні та логічні переходи між полями (при натисканні «Tab»). Додайте кнопку «Пропустити навігацію» для тих, хто використовує читачі тексту.

Висновки

В роботі проаналізовано особливості розробки web-додатків та етапи їх розробки. Також було проаналізовано стан технологій для підвищення доступності у web-розробці. Описано рекомендації для перевірки доступності кінцевого web-додатку.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

1. Адаптація веб-сайтів для людей з вадами зору [Електронний ресурс]. — Режим доступу: <http://cnap.in.ua/wp-content/uploads/2017/08/10.pdf>.

2. Основные этапы разработки веб приложений, сайтов [Електронний ресурс]. — Режим доступу: <http://akonan.ru/index.php?id=15>.

3. Стоит ли учить JavaScript: перспективы, ситуация на рынке труда, мнения экспертов [Електронний ресурс]. — Режим доступу: <https://ru.hexlet.io/blog/posts/stoit-li-uchit-javascript-perspektivy-situatsiya-na-rynke-truda-mneniya-ekspertov>.

4. Библиотека React: особенности, перспективы, ситуация на рынке труда [Електронний ресурс]. — Режим доступу: <https://ru.hexlet.io/blog/posts/biblioteka-react-review-article>.

5. Зачем изучать Node.js, или О перспективах бэкенда на JavaScript: особенности, перспективы, ситуация на рынке труда [Електронний ресурс]. — Режим доступу: <https://ru.hexlet.io/blog/posts/zachem-izuchat-node-js-ili-o-perspektivah-bekenda-na-javascript>.

6. WAI-ARIA: стандарт доступности активных интернет-приложений: особенности, перспективы, ситуация на рынке труда [Електронний ресурс]. — Режим доступу: <https://habr.com/ru/post/40730/>

Захарченко Сергій Михайлович, проф. кафедри обчислювальної техніки, ВНТУ, кафедра обчислювальної техніки, z@vntu.edu.ua.

Zakharchenko Serhii Mikhailovich, prof. Department of Computer Engineering, VNTU, Department of Computer Engineering z@vntu.edu.ua.

Козяр Сергій Олегович, магістр кафедри обчислювальної техніки, ВНТУ, кафедра обчислювальної техніки, catalonya1111@gmail.com.

Kozyar Serhii Olehovych, master of the Department of Computer Engineering, VNTU, Department of Computer Engineering, catalonya1111@gmail.com.