Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)
(рівень вищої освіти)

Спеціалізована клієнт-серверна система на основі технологічного стандарту «WAI-ARIA».
(тема)

Виконав: студент 2 курсу, групи СКС-м-18-2
Литвишко П.В
(прізвище, ініціали)

Спеціальність 123 Комп'ютерна інженерія
(код і повна назва спеціальності)

Тип програми Освітньо професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Спеціалізовані комп'ютерні системи
(повна назва освітньої програми)

Керівник професор кафедри АПОТ
д.т.н Литвинова Є.І
(підпис) (посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ _____
(підпис) (прізвище, ініціали)

2019 р

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
Кафедра Автоматизації проектування обчислювальної техніки
Рівень вищої освіти другий (магістерський)
Спеціальність 123 – Комп'ютерна інженерія
Тип програми Освітньо-професійна
Освітня програма Спеціалізовані комп'ютерні системи

(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Литвишку Павлу Вікторовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Спеціалізована клієнт-серверна система на основі технологічного стандарту «WAI-ARIA»
затверджена наказом по університету від ____ _____20__ р. № _____ 2.
Термін подання студентом роботи до екзаменаційної комісії _24_ __грудня__ 2019 р.
3. Вихідні дані до роботи
Стандарт в області веб розробки «WAI-ARIA»
Мова програмування Javascript
Мова гіпертекстової розмітки HTML
Каскадні таблиці стилів CSS

Синтезатор голосу NonVisual Desktop Access

4. Перелік питань, що потрібно опрацювати в роботі
Дослідження існуючих проблем веб-доступності
Аналіз стандарту в області веб-розробки «WAI-ARIA»
Огляд існуючих асистивних технологій безбар'єрного доступу в мережу для людей з обмеженими можливостями
Проектування і розробка системи у відповідності до стандарту

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів)
15 слайдів

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	03.09.2019 - 10.09.2019	
2	Аналіз предметної області, постановка задачі	10.09.2019 -30.09.2019	
3	Аналіз джерел з проблемної галузі	30.09.2019 -15.10.2019	
4	Дослідження користувацьких сценаріїв	15.10.2019 -15.11.2019	
5	Проектування доступного інтерфейсу	15.11.2019 - 25.11.2019	

6	Розробка сценаріїв програмної реалізації	25.11.2019 -05.12.2019	
7	Розробка веб-сайту	05.12.2019 -15.12.2019	
8	Оформлення пояснювальної записки	15.12.2019 -20.12.2019	
9	Оформлення графічного матеріалу	20.12.2019-22.12.2019	
10	Захист проекту	24.12.2019	

Дата видачі завдання _____

Студент _____
(підпис)

Керівник роботи _____ _____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить 57 сторінок, 9 рисунків, 13 джерел за переліком посилань та 3 додатки.

ДОСТУПНІСТЬ, WAI-ARIA, СТАНДАРТ, СКРИНРИДЕР, HTML, ПРОЕКТУВАННЯ, АСИСТИВНА ТЕХНОЛОГІЯ.

Метою даної роботи є дослідження проектування та розробки спеціалізованого веб-ресурсу, яким зможуть користуватися люди з обмеженими можливостями.

Спроектований і розроблений сайт офтальмологічної клініки, функціоналом якого можуть користуватися всі люди, незалежно від їх зорових

та опорно-рухових обмежень. Для реалізації цього веб-ресурсу був застосований стандарт в області доступної веб-розробки «WAI-ARIA», розглянуті його основні переваги.

Програмна реалізація системи здійснена за допомогою мови Javascript та стандартизованих практик для мови розмітки HTML у середовищі розробки SublimeText 3.

ABSTRACT

The explanatory note contains: 57 pages, 9 figures, 13 links and 3 applications.

ACCESSIBILITY, WAI-ARIA, STANDARD, SCREEN READER, HTML, DESIGN, ASSISTANT TECHNOLOGY.

The purpose of this work is to explore the design and development of a dedicated web resource that can be used by people with disabilities.

The site of the ophthalmology clinic is designed and developed, which functionality can be used by all people, regardless of their visual and musculoskeletal restrictions. The WAI-ARIA accessible web development standard was applied in implementing this web resource and its main advantages are considered.

The system software is implemented using Javascript and standardized practices for HTML markup language in the SublimeText 3 development environment.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

UX – user experience (користувацький досвід)

HTML – HyperText Markup Language (мова гіпертекстової розмітки)

ADA – Americans with Disabilities Act (антидискримінаційний закон по ознаці інвалідності)

API – Application programming interface (програмний інтерфейс додатку)

WAI-ARIA – Web Accessibility Initiative - Accessible Rich Internet Applications (стандарт в області розробки користувацьких інтерфейсів)

XML – eXtensible Markup Language (розширювана мова розмітки)

AJAX – Asynchronous Javascript and XML (асинхронний js та XML)

ВСТУП

Сучасна епоха встановлює соціальну справедливість і рівноправ'я в якості моральних основ суспільства. Принципи поваги людської гідності незалежно від стану фізичного і психічного здоров'я, віку, статі, віросповідання і соціального положення включають дотримання прав людини, в тому числі право на медичне обслуговування, освіту і працю.

Всесвітня павутина - це найважливіше джерело інформації за багатьма соціально-культурним напрямками, таким як: освіта, працевлаштування, управління, торгівля, охорона здоров'я, рекреація, і багато інших. Вкрай важливо, щоб мережа забезпечувала рівний доступ до інформації людям з різними можливостями.

Основна мета Всесвітньої мережі - приносити користь всім людям, незалежно від наявного обладнання або програмного забезпечення, від мови спілкування, місця проживання або можливостей здоров'я. Якщо Інтернет виконує дані умови, він стає доступним для людей з різними слуховими, руховими, зоровими і пізнавальними можливостями. У той же час, якщо вебсайти, онлайн-додатки, мережеві технології або інструменти спроектовані невдало, вони можуть створювати для людей перепони на шляху до використання Всесвітньої мережі.

У доступності зацікавлені не тільки люди з обмеженими можливостями. Зір часто погіршується з віком, що ускладнює читання дрібного тексту. Спритність пальців у людей різного віку також різна, що ускладнює натискання на інтерактивні елементи.

Властивість Інтернет-ресурсу, що враховує всі можливі проблеми, з якими може зіткнутися людина при його використанні, називається вебдоступністю. При створенні по-справжньому якісного сайту, питання його доступності для всіх категорій користувачів, незалежно від їх фізичних можливостей і обмежень, повинно бути одним із першочергових завдань. Концепція доступності розглядає всіх людей рівноправними членами суспільства при будь-яких умовах.

Одним з важливих кроків при створенні додатків є розуміння різноманітності ваших користувачів. Є люди, які обмежені в своїх можливостях через різні проблем з візуальним сприйняттям, моторикою, слухом, вимовою або сприйняттям інформації. Через це їх способи взаємодії з вашим додатком відрізняються. Багато з них для роботи з вашим додатком використовують голосове читання з екрану, пристрої Брайля або відстеження напрямку погляду. Такі обмеження можливостей користувачів можуть бути тимчасовими або постійними.

Використання на вебсайтах складних для користувача інтерфейсів постійно зростає. Для того, щоб робота з такими інтерфейсами була доступна людям з обмеженими можливостями, які користуються спеціальними допоміжними технологіями, має бути забезпечено їх коректна взаємодія з елементами управління сайту. Для надання можливості повноцінного використання Інтернету людям з фізичними обмеженнями були розроблені технологічні стандарти, які висувають вимоги до розроблюваних елементів інтерфейсу та розмітки сторінок [1].

Ця робота присвячена розробці спеціалізованої клієнт-серверної системи на основі технологічного стандарту «WAI-ARIA». Щоб досягти цієї мети ми повинні зробити наступне:

а) проаналізувати існуючі стандарти розробки доступних сайтів;

б) проаналізувати існуючі проблеми з якими стикаються користувачі з фізичними обмеженнями, або які використовують допоміжні апаратні чи програмні засоби при взаємодії з сайтами в мережі Інтернет;

в) показати можливість використання «WAI-ARIA» для вирішення цих проблем, а також покращення користувацького досвіду веб-ресурсу.

1 ТЕОРЕТИЧНА ЧАСТИНА

1.1 Актуальність дослідження

У 2018-2019 роках в Україні відбулося різке зростання Інтернет проникнення. Показник відсоткової кількості Інтернет-користувачів від загальної кількості населення країни зріс на майже 8%. Основними передумовами такого росту стали доступність смартфонів та активне розгортання третього та четвертого поколінь мобільного зв’язку, які дали

можливість селам та невеликим містам користуватися швидкісним Інтернетом. Рівень активних користувачів в таких населених пунктах впевнено наближається до показників великих міст. Продовжується вирівнювання структури абонентів пропорційно до структури населення. Цей процес вже майже завершений по належності людини до конкретної статі, і триває по інших показниках: територіальному, віковому, професійному, освітньому, рівню доходів. Майже все населення України віком до 35 років є активними Інтернет-користувачами.

Це підтверджує те, що Інтернет став таким глобальним і розповсюдженим явищем, що він зачіпає всі аспекти нашого життя і є джерелом інформації для мільйонів людей незалежно від їх соціального, матеріального, а також стану їх здоров’я. Саме тому сьогодні вкрай важливо приділяти увагу розробці безбар’єрних Інтернет ресурсів, яким зможуть користуватися абсолютно всі без винятку, хто в цьому зацікавлений. Мережа об'єднує людей по всьому світу, допомагає нам отримувати новини та інформацію, зберігати документи, а також планувати маршрут подорожей і навіть зберігати своє здоров'я. Проте, не всі сайти дбають про доступність вебконтенту. Існує багато сайтів, недоступних для людей з обмеженими можливостями.

Близько 15% людей на Землі мають певні вади слуху, зору, руху або когнітивних здібностей. В Україні проживає 240 тисяч сліпих і слабозрячих. Всього в світі налічується близько 314 мільйонів людей, що мають порушення зору, викликані різними причинами, 45 мільйонів з них є сліпими. Вони користуються Інтернетом за допомогою програм читання з екрану, екранних луп і клавіатури. Доступ до інформації для таких людей включений в список прав людини, але деякі вебсайти для них недоступні. Щоб їм було зручно користуватися сайтами, потрібно по-особливому оформляти розмітку, ілюстрації, форми введення, таблиці та інші елементи сторінки.

В даний час більшість вебсайтів недоступні для людей з інвалідністю, в тому числі сайти, призначені для надання держпослуг, і найбільш популярні приватні веб-ресурси. У більшості ситуацій недоступність сайту - штучна перешкода, яку легко і без істотних додаткових витрат можна було б обійти на стадії розробки веб-ресурсу. В цілому для забезпечення доступності сучасних вебсайтів в більшості випадків достатньо всього лише, щоб веб-дизайнер і контент-менеджери дотримувалися вимог керівництва по забезпеченню доступності веб-ресурсів, розробленого світовим веб-консорціумом «WAIARIA» [2].

В останні два десятиліття для забезпечення доступності телекомунікацій було зроблено чимало зусиль, що спиралися в основному на підтримку організацій, які виступають за права інвалідів. Рубіжною подією в цьому відношенні стало прийняття Конвенції ООН про права інвалідів, учасниками якої в даний час є 153 держави, включаючи Україну. Цей документ формулює законодавчі рамки, такі необхідні для вироблення єдиних глобальних стандартів і політичних рішень.

Для більшості цих користувачів інформація та послуги в тому вигляді, в якому вони зараз представлені в мережі, є недоступними частково або повністю через їх неправильну подачу і оформлення. Для того, щоб залучити цю велику категорію громадян в інформаційні та господарські процеси за допомогою мережі Інтернет недостатньо надання комп'ютерів і виходу в мережу. Необхідна спеціальна організація контенту, яка зробить його доступним.

Таким чином, основна проблема інтерфейсів додатків для людей з вадами зору в тому, що часто розробники не думають про доступність взагалі. Велика кількість команд розробників та дизайнерів не визнають технологічні інновації в сфері асистивних технологій, правові і значні демографічні зміни, тому що вони розглядають доступність як обмежувальну і необов’язкову

частину створення сайту з гарним UX для всіх користувачів. Процес дизайну сьогодні зводиться до проектування нових сценаріїв взаємодії з користувачем. Ми можемо спроектувати цікаву форму, захоплюючий інтерфейс, який буде зручно використовувати, але при непродуманій логіці, ланцюжку взаємодій і, найголовніше, зрозумілої форми взаємодії з тим, хто з цим інтерфейсом працює, система не буде користуватися попитом.

Важливо розуміти, що веб-доступність стосується не тільки громадян, які мають інвалідність, а набагато більш широкої групи населення, що включає як осіб з обмеженими можливостями здоров'я, в т.ч. і з інвалідністю по зору, слуху, з обмеженою моторикою, так і людей з обмеженою мобільністю, в т.ч. літніх людей, в яких погіршився зір в зв’язку з віковими змінами, так і користувачів з технічними обмеженнями (користувачі мобільних пристроїв, відсутність широкополосного виходу в мережу Інтернет та інше).

Щоб забезпечити певний рівень доступності, сайт повинен бути спроектований і написаний у відповідності до сучасних технологічних стандартів в області розробки доступної веб інфраструктури. Одним з таких є «WAI-ARIA» - технологічний стандарт, що розробляється Консорціумом Всесвітньої павутини для надання можливості повноцінного використання Інтернету людьми з фізичними обмеженнями (порушення роботи органів зору та опорно-рухового апарату).

Забезпечення доступності Інтернет-простору вигідно для окремих громадян, комерційних підприємств і суспільства в цілому. Міжнародні стандарти визначають необхідні умови для реалізації веб-доступності.

1.2 Визначення доступності вебсайту

Доступність вебсайтів - це всеосяжна практика, що забезпечує відсутність бар'єрів, що перешкоджають взаємодії або доступу до вебсайтів у всесвітній павутині людьми з фізичними вадами, ситуаційними порушеннями

і соціально-економічними обмеженнями пропускної здатності і швидкості Інтернету. Коли сайти правильно спроектовані, розроблені і відредаговані, як правило, всі користувачі мають рівний доступ до інформації і функціональності.

В 2012 році управління США по громадянським правам дало визначення терміну наступним чином: «доступний» означає, що люди з обмеженими можливостями можуть самостійно отримувати ту ж саму інформацію, брати участь в тих же взаємодіях і користуватися тими ж послугами протягом того ж періоду часу, що й особи без інвалідності, з практично еквівалентною простотою використання.

Наприклад, коли сайт кодується за допомогою семантично значущого HTML, з текстовими еквівалентами, наданими для зображень, і з змістовно названими посиланнями, це допомагає сліпим користувачам використовувати програмне забезпечення перетворення тексту в мову або апаратне забезпечення перетворення тексту в Брайль. Коли текст і зображення великі або збільшуються, користувачам з поганим зором легше читати і розуміти вміст. Коли посилання підкреслені (або іншим чином диференційовані), а також пофарбовані, це гарантує, що люди з кольоровою сліпотою зможуть їх помітити. Коли клікабельні посилання і області великі, це допомагає користувачам, які не можуть точно управляти мишею. Коли сторінки не закодовані таким чином, який ускладнює навігацію за допомогою однієї клавіатури, це допомагає користувачам, які не можуть використовувати мишу або навіть стандартну клавіатуру. Коли відео має субтитри або має супровід на мові жестів, це можуть зрозуміти глухі і слабочуючі користувачі. Коли миготливі ефекти виключаються або стають необов'язковими, користувачі, схильні до нападів, викликаним цими ефектами, не піддаються ризику. А коли зміст написаний простою мовою і проілюстрований навчальними схемами і анімацією, користувачі з дислексією і труднощами в навчанні краще засвоюють матеріал. Коли сайти правильно побудовані і підтримуються, всі ці

користувачі можуть бути розміщені без зниження зручності використання сайту для звичайних користувачів.

1.3 Підстави для впровадження доступності вебсайтів

1.3.1 Правовий аспект

У 1973 році конгресом США був прийнятий закон "American Workforce Rehabilitation Act", який був покликаний усунути дискримінацію інвалідів в сфері праці в державних структурах, а також структурах, які отримують федеральне фінансування. Закон складається з чотирьох основних розділів: 501, 503, 504 і 508. Останній розділ (Section 508) " Стандарт доступності електронних та інформаційних технологій". Тобто, якщо ви хочете продати будь-який продукт федеральному органу США, то ваш сайт мусить відповідати вимогам розділу 508. Цей закон був прийнятий в 1998 році і був покликаний реалізувати антидискримінаційні заходи в сфері електронних та інформаційних технологій, таких як комп'ютерна техніка, мережа Інтернет і мультимедійна продукція.

Крім того, Розділ III Закону про американців-інвалідів (ADA) забороняє дискримінацію за ознакою інвалідності. Багато судів і Міністерство юстиції США зайняли позицію, згідно з якою ADA вимагає від операторів та власників вебсайтів і додатків прийняття позитивних заходів, щоб зробити їх вебсайти і додатки доступними для інвалідів і сумісними з загальними допоміжними технологіями, такими як програма читання з екрану JAWS. Міністерство юстиції США схвалило стандарт WAI-ARIA як відповідний стандарт доступності в декількох розрахункових угодах.

У США, крім федеральних, штатних і місцевих урядових веб-сайтів, які повинні відповідати правилам Розділу 508, не існує обов'язкових правових стандартів ADA, яких необхідно дотримуватися для забезпечення доступності веб-сайтів. Однак той факт, що не існує простого набору правових вимог для

доступності веб-сайту, не означає, що вашому бізнесу не буде пред'явлений позов. У 2017 році в США було порушено близько 800 справ, пов'язаних з доступністю в Інтернеті, і в 2018 році - понад 2200. Крім того, хоча в 2010 році Міністерство юстиції заявило, що опублікує рекомендації з питання забезпечення доступності в Інтернеті, в 2017 році цей план був скасований , що також стимулювало судові позови проти недоступних сайтів.

Подібні закони є в ряді країн Європейського союзу, а також Норвегії, Ізраїлю, Японії, Канади, Австралії, Філіппін, Бразилії та Росії. Переважно ці закони стосуються державних ресурсів та мають рекомендаційний характер для інших сайтів, базуючись на технологічних стандартах WAI-ARIA або WCAG 2.0 [3].

Станом на 2019 рік в Україні будь-яке юридичне регулювання цієї сфери відсутнє.

1.3.2 Фінансовий аспект

Компанії, які охоплюють людей з порушеннями, тільки виграють від цього. Спроби зробити веб-сайт або додаток повністю доступними позитивно впливають на репутацію і створюють образ соціальної відповідальності і турботи про всіх користувачів. Люди з обмеженими технічними або фізичними можливостями – це також клієнти, які купують товари чи користуються послугами, тому забезпечення доступності також сприяє розширенню клієнтського списку та виходу на новий ринок.

Імплементація доступності на проекті не завжди є легкою справою і, ймовірно, збільшить час, витрачений на етапи розробки. Однак це скоротить час, витрачений на вдосконалення сайту та його технічну підтримку. Думка про доступність з самого початку цифрового проекту також допоможе зменшити подальші витрати, включаючи обслуговування та підтримку клієнтів.

Більше 80% людей вирішують не довіряти постачальнику послуг через бар'єри, одна з головних причин - погана доступність Інтернету. Оскільки більшість брендів їх часто ігнорують, користувачі з обмеженими можливостями готові витратити більше своїх грошей на компанії, які надають їм продуманий досвід користувачів. Купівельна спроможність саме цього ринку величезна.

У процесі додавання функцій доступності гіганти Кремнієвої долини не соромляться впроваджувати інновації, вкладаючи мільйони в нові технології та новаторські функції. Наприклад, Facebook зараз використовує штучний інтелект для надання своїм сліпим користувачам автоматичного альтернативного тексту за допомогою технології розпізнавання об'єктів. Ця функція тепер дозволяє їм повною мірою насолоджуватися 2 мільярдами фотографій, якими щодня діляться у Facebook. Ідея здебільшого виникла від Метта Кінга, сліпого програмного інженера, який працює з командою з доступності. Facebook виявив прихильність бути доброзичливою компанією для людей з обмеженими можливостями як в Інтернеті, так і в фізичному світі.

З точки зору SEO, доступний веб-сайт також допоможе отримати кращі рейтинги. Пошукові системи, такі як Google, діють так само, як сліпі люди, що використовують зчитувач екрана. Під час сканування сторінок роботи розуміють лише текст, тому важливо спланувати архітектуру ресурсу. Компанія Google, яка займає лідируючу позицію на ринку пошукових систем, враховує доступність сайту під час ранжирування пошукової видачі. Це, в свою чергу, напряму впливає на кількість відвідувачів вашого сайту та його фінансові показники. Для тестування сайтів на доступність в браузері Google Chrome в інструментах розробника (Chrome DevTools) є вбудована функція аудиту інтерфейсів (рисунок 1.1).

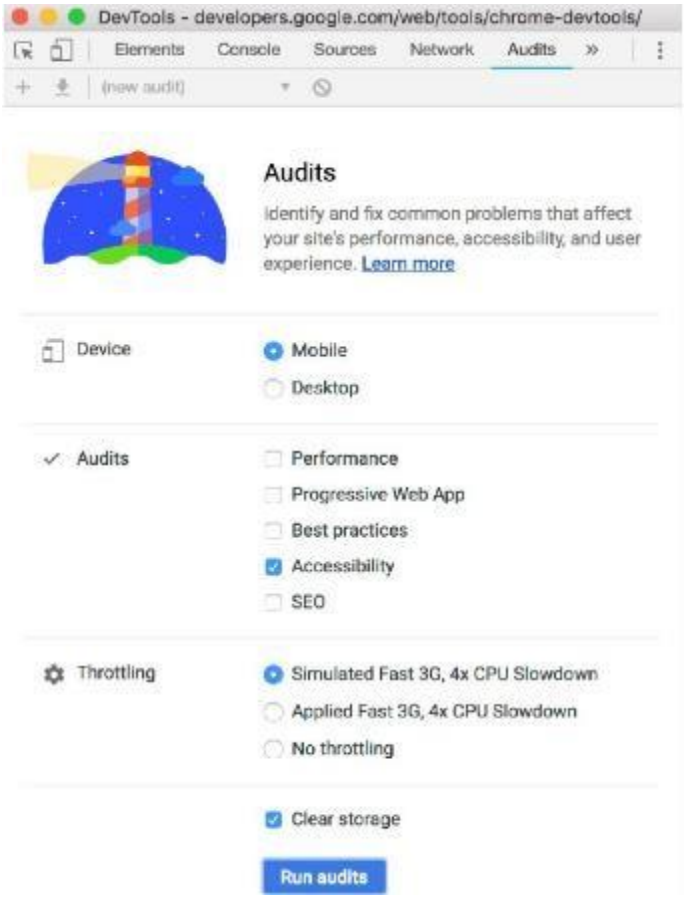

Рисунок 1.1 – Тестування доступності в Google Chrome

Покращення доступності не відверне поточних клієнтів, але приверне нових клієнтів та покращить конверсії. Визнаючи користувачів з порушеннями, компанії дають їм знати, що їхні права мають значення і що їх бізнес цінується.

1.3.3 Моральний аспект

Доступність в інтернеті має важливе значення для рівноправного і справедливого суспільства. Чим більше наша соціальна і професійна сфера життя переходить в онлайн, тим важливішою для людей з обмеженими можливостями стає здатність брати участь у всіх онлайн взаємодіях без додаткових бар'єрів. Подібно до того, як архітектори будівель можуть піклуватися або нехтувати технологіями доступності, такими як рампи для

інвалідних колясок, веб-розробники можуть допомагати або заважати користувачам, які використовують додаткові інструменти для виходу в інтернет.

Найчастіше покращення доступності сайту приносить користь всім. Хоча ми зазвичай думаємо про людей з обмеженими можливостями, як про людей, які перебувають в цьому стані постійно, насправді будь-хто може отримати тимчасову втрату функціональності. Наприклад, хтось є сліпою людиною, у когось тимчасова очна інфекція, а хтось просто перебуває на вулиці під яскравим сліпучим сонцем. Все це може бути причиною, чому користувач в даний момент не здатний побачити, що відображено на екрані. Будь-хто може з якоїсь причини на час стати обмеженим у можливостях, і тому поліпшення доступності ваших веб-сайтів поліпшить зручність всіх користувачів, незалежно від їх стану.

Інтернет – одна з найкращих речей, яка коли-небудь траплялася з людьми з обмеженими можливостями. Можливо, не всі розробники думали про це таким чином, але все що їм потрібно зробити, так це згадати дні до Інтернету, щоб зрозуміти, чому це так. Наприклад, як сліпі люди читали газети до появи Інтернету? В основному вони цього не робили. Аудіокасети або роздруківки Брайля були надто дорогими, брайлевська версія The New York Times була надто громіздкою щоб бути практичною. У кращому випадку вони могли попросити членів сім’ї чи інших прочитати їм газету. Цей метод працює, але він робить людей залежними від інших.

Більшість газет в даний час публікує свій контент в Інтернеті у форматі, який потенційно може бути прочитаний програмами читання з екрану сліпими людьми. Програми читають вслух електронний текст, щоб сліпі могли користуватися комп'ютерами та отримувати доступ до будь-якого текстового контенту за допомогою комп'ютера. Неочікувано сліпі люди не повинні покладатися на інших, щоб вони прочитали їм газету. Їм не потрібно чекати дорогих аудіокасет або дорогих громіздких шрифтів Брайля. Вони просто

відкривають веб-браузер і слухають, як їх програма читання з екрану читає їм газету, і вони роблять це незалежно, коли хочуть, і як тільки контент публікується.

Точно так само люди з руховими порушеннями, які не можуть взяти газету або перевернути її сторінки, можуть отримати доступ до інтернет-газет через свій комп'ютер, використовуючи певні допоміжні технології, які адаптують інтерфейс комп'ютера до їх власних порушень. Іноді адаптації прості, наприклад, коли людина кладе указку в рот і використовує її для введення команд клавіатури. В інших випадках адаптація є більш складною, як при використанні спеціальних клавіатур або програмного забезпечення для відстеження руху очей, яке дозволяє людям використовувати комп'ютер не більше ніж рухами очей.

Люди, які є глухими, можуть читати газети самостійно, але вони також можуть читати стенограми онлайн або заголовки мультимедійного контенту онлайн. Багато людей з когнітивними порушеннями також можуть отримати велику користь із структури і гнучкості веб-контенту.

Кожна з основних категорій інвалідності вимагає певних типів адаптації в дизайні веб-контенту. У більшості випадків така адаптація приносить користь практично всім, а не тільки людям з обмеженими можливостями. Майже всі отримують користь від корисних ілюстрацій, правильно організованого контенту і зрозумілої навігації. хоча субтитри є необхідністю для глухих користувачів, вони можуть бути корисні для інших, включаючи будь-якого, хто переглядає відео без звуку.

1.4 Огляд допоміжних програмних і апаратних технологій

Програма читання з екрану - програма, що дозволяє визначати і інтерпретувати (зазвичай у вигляді звукових і / або голосових) сповіщень події, що відбуваються на екрані. Програми читання з екрану добре забезпечують

доступ до трьох найбільш широко використовуваних дій на персональних комп'ютерах, обробка текстів, перегляд інтернету та електронна пошта. Цей тип технології підходить не тільки для сліпих людей, а й для людей, які мають проблеми з читанням тексту, який відображається на цифровому екрані [4]. На сьогоднішній день у користувачів є кілька програм читання з екрану, з яких вони можуть вибирати.

NVDA - це програма невізуального доступу до робочого столу комп'ютера, що працює за рахунок мовного оповіщення користувача про об'єкти на робочому столі і вікнах, діях і процесах. Основною метою проекту NVDA є надання можливості незрячим працювати за комп'ютером, не переплачуючи за спеціалізоване програмне забезпечення, так як поширюється абсолютно безкоштовно. Оскільки NVDA є проектом з відкритим вихідним кодом, кожен користувач, при наявності знань достатнього рівня, зможе внести свій внесок в розвиток програми або зробити її зручніше для себе. Завдяки спільноті, NVDA перекладена більш, ніж на 20 мов, серед яких і українська. Всі мови входять в дистрибутив, користувачеві не треба качати окрему версію для кожної мови. NVDA складається з різних підсистем, включаючи менеджер надбудов, модулі додатків, обробник подій і обробники введення і виведення, а також модулі для підтримки API доступу. NVDA використовує об'єкти для представлення об'єктів у програмі, таких як рядки меню, рядка стану і різні вікна переднього плану. NVDA збирає різну інформацію про об'єкт, таку як його ім'я, значення і координати екрану, через API спеціальних можливостей, що надаються об'єктом.

Програмне забезпечення підтримує скорописний, нескорописний та комп’ютерний Брайль для деяких мов. NVDA використовує унікальну систему індикації координат миші за допомогою аудіо сигналів, а також надає можливість для обробки подій, таких як натискання клавіш, зміна імені та коли додаток отримує або втрачає фокус. Також при наведенні миші на об'єкт він озвучується. NVDA повністю портабельна, може працювати зі зйомних носіїв

(в тому числі і компакт-дисків) без функціональних обмежень, не залишаючи слідів в системі.

Інша асистивна технологія AbleNet присвячена тому, щоб змінити життя людей з обмеженими можливостями шляхом створення допоміжних технологічних продуктів, які перетворюють їхнє життя простими, відчутними та потужними способами. Цю технологію використовує такий тип девайсів як Switch (рисунок 1.2). Перемикачі такого типу мають одну або декілька великих кнопок в одному пристрої, на які користувач може призначити дії для різних потреб, таких як прокрутка змісту сторінки чи вибору інтерактивного елементу.

Рисунок 1.2 – Blue 2 Bluetooth Switch

Доступність - це не тільки про комп'ютери та ноутбуки, це ще й про мобільні пристрої. Людина з обмеженими можливостями може перейти на свій веб-сайт із телефону або планшета. Цей пристрій підключається до iOS або Androidдевайсу, і на його кнопки можна назначити якісь призначені для

користувача дії. Наприклад, цей Switch - з двома великими кнопками, зручно призначити на нього управління фокусом на сторінці.

Для користувачів, які не бачать ваш інтерфейс, синтез мови буває зручний не завжди. Тому існує брайлівський дисплей (рисунок 1.3). Це спеціальний пристрій виведення, який відображає інформацію у вигляді відчутних точок з абетки Брайля. Пристрій перетворює візуальну інформацію на набір знаків, які складаються з шести точок, які можна тактильно відрізнити пальцями рук. Але у нього є два недоліки. Він дуже дорогий, далеко не кожен, хто потребує, може її собі дозволити. Крім того, люди, які втратили зір в більш зрілому віці, часто не знають абетку Брайля і не хочуть її вчити.

Рисунок 1.3 – Брайлівський дисплей

Програмних технологій теж достатньо. Це інструменти для збільшення зображення на сторінці, наприклад, екранні лупи, які вбудовані в Windows, macOS, це програмні модифікатори колірної гами та інше. Існує програмне

забезпечення, яке дозволяє керувати інтерфейсом за допомогою руху очей і голови. У macOS така програма вбудована, вона називається Dwell Control.

1.5 Засоби розробки

1.5.1 HTML

Мова розмітки гіпертексту (Hypertext Markup Language), або, як її частіше називають, HTML, - це комп'ютерна мова, що лежить в основі World Wide Web (Всесвітньої Павутини). Завдяки мові HTML будь-який текст можна розмітити, перетворивши його в гіпертекст з подальшою публікацією в інтернет.

Мова HTML має власний набір символів, за допомогою яких Webбраузери відображають сторінку. Ці символи, звані дескрипторами, включають в себе елементи, необхідні для створення гіперпосилань.

Однією з відмінних рис HTML-документів є те, що сам документ містить лише текст, а всі інші об'єкти вбудовуються в документ в момент його відображення браузером за допомогою спеціальних тегів і зберігаються окремо. При збереженні HTML-файлу в місці розміщення документа створюється папка, в яку поміщаються супроводжуючі його графічні елементи оформлення

1.5.2 CSS

Під терміном CSS ми розуміємо фразу "Каскадні таблиці стилів". Вони впроваджуються в HTML-код і не вимагають ніяких спеціальних редакторів і компіляторів, адже інтерпретуються вони звичайним браузером.

CSS використовується розробниками веб-сторінок для завдання кольорів, шрифтів, розташування окремих блоків і інших аспектів представлення зовнішнього вигляду цих веб-сторінок. Основною метою

розробки CSS було відділення опису логічної структури веб-сторінки (яке проводиться за допомогою HTML або інших мов розмітки) від опису зовнішнього вигляду цієї веб-сторінки, який тепер проводиться за допомогою формальної мови CSS. Такий поділ може збільшити доступність документа, надати велику гнучкість і можливість управління його поданням, а також зменшити складність і повторюваність в структурному вмісті.

Крім того, CSS дозволяє представляти один і той же документ в різних стилях або методах виведення, таких як екранне представлення, друковане подання, читання голосом (спеціальним голосовим браузером або програмою читання з екрану), або при виведенні пристроями, що використовують шрифт Брайля.

1.5.3 Bootstrap framework

Bootstrap також відомий як Twitter Bootstrap- вільний набір інструментів для створення сайтів і веб-додатків. включає в себе HTML і CSS шаблони оформлення для типографіки, веб-форм, кнопок, міток, блоків навігації та інших компонентів веб-інтерфейсу, включаючи JavaScript-розширення.

Bootstrap використовує найсучасніші напрацювання в області CSS і HTML, тому часто необхідно бути уважним при підтримці старих браузерів які не підтримують деякі функції.

Основною перевагою bootstrap є економія часу при розробці за рахунок використання шаблонів і класів дизайну.

Динамічні макети Bootstrap масштабируются на різні пристрої і резалюції екрану без будь-яких змін в розмітці, що скорочує час оптимізації проекту під різні розміри екрану.

Використовуються єдиний стиль і шаблони за допомогою центральної бібліотеки. Дизайн і макети веб-сторінок узгоджуються один з одним. Простота у використанні платформи дозволяє, користувачеві з базовими

знаннями HTML і CSS розробляти шаблони сайтів професійно і в найкоротші терміни.

Сумісність з браузерами в Twitter Bootstrap також є безперечною перевагою. Bootstrap коректно відображається на останніх версіях веббраузерів Mozilla Firefox, Yandex Browser, Google Chrome, Safari, Internet Explorer, Microsoft Edge і Opera;

Особливістю Twitter Bootstrap є відкритий вихідних код і безкоштовне розповсюдження.

1.5.4 Javascript

JavaScript можна назвати однією з найбільш часто використовуваних прототипно-орієнтованих, скриптованих мов програмування. Javascript код може виконуватися як на стороні клієнта, так і на стороні серверу. Практично жоден з сучасних сайтів не обходиться без використання даної мови програмування. JavaScript був розроблений компанією Netscape для надання інтерактивності веб-сторінкам.

Найбільш популярні області використання javascript:

– створення динамічних HTML сторінок;

– валідація даних, які ввів користувач до передачі їх на сервер;

– динамічне створення документу за домогою сценарію;

– взаємодія з користувачем задля вирішення локальних задач.

1.6 Технологічний стандарт «WAI-ARIA»

WAI-ARIA - стандарт доступності активних Інтернет-додатків, визначає підходи до того, щоб зробити вміст сайтів та інтернет-додатки більш доступними для людей з обмеженими можливостями. Стандарт описує критерії доступності для різних категорій користувачів. На практиці для

дотримання високого рівня відповідності критеріям, тестувальнику буде корисно уявити себе на місці наступних категорій користувачів (або залучити їх до тестування):

1) страждаючі різними видами дальтонізму (кольорової сліпоти). Там, де колір використовується для індикації або надання інформації, повинні бути передбачені альтернативні візуальні засоби;

2) слабозорі. Накладаються вимоги по контрастності, розмірам елементів і підтримки масштабування сторінки;

3) сліпі (повністю незрячі користувачі екранних зчитувачів). Потрібен широкий забір заходів, наприклад, надання текстової альтернативи для всіх значущих нетекстових елементів, написання семантичної верстки, правильне використання семантичних областей, заголовків і інших навігаційних елементів. Проектування додаткових засобів навігації по сторінці, наявність додаткової метаінформації про елементи на сторінці і зв'язках між ними. Обов'язкове надання текстових міток і, при необхідності, підказок для елементів текстового вводу. Потрібно врахувати особливості сприйняття вмісту, воно сприймається на слух, тобто послідовно, без можливості охоплення всієї сторінки поглядом цілком, без можливості помітити інформацію в іншій області сторінки (якщо вона не пов'язана відповідним чином), без сприйняття сенсорних характеристик тексту. Необхідно реалізовувати взаємодії з елементами управління або введення за допомогою набору атрибутів розмітки;

4) користувачі з порушенням слуху. Потрібно обов'язкове надання текстової альтернативи для аудіоконтенту;

5) користувачі з порушенням опорно-рухового апарату, яке може виражатися в нездатності користуватися мишею. Потрібна повна керованість і доступність сайту за допомогою клавіатури.

Застосування стандарту особливо ефективно для сайтів з динамічним вмістом і розвиненими елементами управління користувацького інтерфейсу, розробленими з використанням Ajax, HTML, JavaScript, і пов'язаних з ними технологіями [5]. В даний час, деякі функціональні можливості веб-сайтів недоступні для частини користувачів з обмеженими фізичними можливостями, особливо для людей, які покладаються на програми, які читають з екрану і для тих людей, які не можуть використовувати мишу. WAIARIA задає нові шляхи надання функціональних можливостей програмамипомічниками. За допомогою WAI-ARIA розробники можуть створювати прогресивні веб-додатки, доступні та зручні для всіх користувачів. ARIA - це набір спеціальних атрибутів, які можуть бути додані в будь-яку розмітку, але особливо підходять для HTML.

Все більше сайтів в Інтернеті використовують для управління сторінкою складні і просунуті системи, такі як, деревовидні меню на Javascript. Щоб забезпечити доступ людям з фізичними порушеннями, спеціальним браузерам і технологіям-помічникам необхідно взаємодіяти з цими системами управління, проте інформація, яка для цього потрібна, виявляється недоступною на більшості існуючих сьогодні сайтів.

Розглянемо принцип роботи стандарту WAI-ARIA більш конкретно. Стандарт WAI-ARIA задає набір атрибутів HTML, що дозволяють ідентифікувати взаємодіючі з користувачем фукції сайту, встановити їх взаємозв'язок і поточне розташування. Стандарт описує нові техніки навігації для виділення областей і основних елементів структури сайту, таких як меню, основний вміст, другорядний вміст, інформація про банери. Наприклад, з використанням WAI-ARIA розробники можуть позначити області на сторінках, дозволяючи користувачам легко переміщатися між областями за допомогою клавіатури без необхідності багаторазово натискати клавішу Tab [7].

WAI-ARIA включає в себе технології для позначення елементів управління, активних областей AJAX, подій і саморобних елементів управління. Техніки стандарту можуть бути використані для кнопок, випадаючих списків, деревовидних елементів, календарів та іншого.

Стандарт WAI-ARIA пропонує авторам сайтів:

– ролі для описання типу активного елементу, наприклад, alert, article, bunner, tab, navigation та ін;

– атрибути для описання стану інтерактивних елементів управління, наприклад, current, hidden, readonly, selected тощо;

– ролі для описання структури сторінок, наприклад, статті, списки, зображення тощо;

– атрибути для задання активних областей сторінки, яка може оновлювати зміст, наприклад, біржові котирування, а також способу появи таких оновлень. Наприклад, важливі оновлення можуть представлятися в діалоговому вікні, а другорядні - з'являтися прямо на сторінці;

– атрибути для перетаскування об’єктів, які описують переміщувані структури і цільові положення в документі. Можливість управління з клавіатури об’єктами і подіями. Приклад використання aria атрибутів для саморобного інтерактивного елементу (рисунок 1.4).

```
<span role="checkbox" aria-checked="mixed"
onkeydown="return checkBoxEvent(event);"
onclick="return checkBoxEvent(event);" >
A checkbox label
</span>
```

Рисунок 1.4 □ Використання aria атрибутів для надання ролі

Іншим прикладом бар'єру на шляху до нормального використання можливостей сайту може служити техніка перетягування об'єктів (drag & drop), недоступна користувачам, які не здатні управляти мишею і мають в розпорядженні тільки клавіатуру. Навіть відносно прості сайти можуть виявитися складними у використанні, якщо вони вимагають величезної кількості натискань клавіш при управлінні за допомогою тільки однієї клавіатури.

Велика кількість інтернет-додатків, розроблених із застосуванням AJAX, DHTML і інших технологій створюють додаткові проблеми доступу до інформації. Наприклад, якщо вміст сторінки змінюється у відповідь на дії користувача, після закінчення певного часу або в зв'язку з раптовими оновленнями, то цей нове вміст може виявитися недоступним для людей, що використовують програми читання з екрану - сліпих або людей з когнітивними розладами.

Стандарт WAI-ARIA дозволяє вирішити ці проблеми доступу, встановлюючи способи зробити всю необхідну інформацію про роботу сайту доступною всім програмам-помічникам. За допомогою стандарту навіть складні інтернет-додатки можна зробити доступними і зручними для людей з обмеженими можливостями.

1.7 Приведення сайту у відповідність до стандарту «WAI ARIA»

Мета стандарту - зробити вміст вебсайту більш доступним для всіх користувачів, незалежно від типу використовуваного браузера (традиційний браузер, голосовий браузер, мобільний телефон та ін.), в тому числі для користувачів з обмеженими фізичними можливостями. Основу доступності складають чотири принципи:

– сприйнятність - інформація і компоненти користувацького інтерфейсу повинні бути представлені тільки в тому вигляді, який можуть сприймати користувачі;

– керованість - компоненти користувацького інтерфейсу і навігації повинні бути керованими;

– зрозумілість - інформація і оперції користувацього інтерфейсу повинні бути зрозумілими кожній людині;

– надійність - зміст повинен бути надійним в такій мірі, яка потребується для його інтерпретації широким колом різних користувацьких додатків, включаючи асистивні технології.

Кожен принцип підтверджується трьома рівнями відповідності (A, AA і AAA), які відображають на якому рівні був дотриманий стандарт. WAI-ARIA широко використовується для розуміння і оцінки інтерфейсу вебдизайнерами, веб-розробниками, тестувальниками, менеджерами проектів та аналітиками.

Далі розглянемо практичні рекомендації, засновані на принципах, описаних в стандарті, які допоможуть створити інклюзивний інтерфейс.

Звуковий і візуальний вміст має мати альтернативне представлення. Необхідно супроводжувати текстовими еквівалентами всі нетекстові елементи, в тому числі img, map, applet, object, frame, script, звукові і відеофайли. Поки браузери не навчаться вимовляти текстовий еквівалент візуального ряду, надавайте аудіо-еквівалент візуального ряду або мультимедіа презентації. Бажано синхронізувати текстовий або аудіо еквівалент з відео по часу. У разі, якщо використовуваний формат відео не підтримує субтитри, можна зробіти два варіанти відео – з субтитрами і без.

Не варто розраховувати тільки на колір. Слід переконатися, що вся інформація, яка передана за допомогою кольору, також доступна без кольорів, наприклад, у формі тексту або розмітки. Кольори переднього плану і фону

повинні бути досить контрастними при перегляді на чорно-білому дисплеї, а також людьми з обмеженим колірним сприйняттям.

Необхідно правильно використовувати розмітку і сторінки стилів. Якщо існує відповідний засіб розмітки, варто використати його, а не зображення для передачі інформації. Щоб відповідати цьому правилу, слід регулярно перевіряти сторінки сайту на відповідність стандартам HTML і CSS за допомогою валідаторів. Створені документи мають відповідати формальним вимогам по формату, допускається розміщення в початку документу посилання для простої перевірки коректності формату документу. У підключених стильових сторінках краще використовувати відносні, а не абсолютні величини в CSS-атрибутах. Наприклад, можна використати em або значення у відсотках замість pt або cm. Якщо використані абсолютні одиниці, необхідне детальне тестування вмісту в різних призначених для користувача середовищах. Для відображення структури документу необхідне обов’язкове використання заголовків. Списки елементів мають бути оформленими за допомогою тегів ol, ul, I dl. Коректно оформлені цитати допомагають зрозуміти користувачу відношення між текстом цитати і її автором, не варто використовувати елементи q і blackquote для оформлення тексту, який не є цитатою.

Мова тексту має бути коректно задекларована. Якщо в тексті використовується не одна мова, то це слід відмічати в місцях її зміни, щоб синтезатори голосу використовували відповідний мовний пакет. Наприклад в HTML використовується атрибут lang, в XML - xml: lang. Якщо документ містить частини тексту на інших мовах, явне вказання мови дозволить звуковому браузеру переключитися на іншу мову при читанні даних фрагментів. При першому вживанні в тексті кожній абривіатурі має бути приведений розгорнутий еквівалент. В HTML для цього використовується

атрибут title для елементів abbr і acronym. Основна мова документу позначається за допомогою атрибуту lang тега meta.

Створення доступних таблиць потребує виділення заголовків рядків і стовпців в таблицях з даними, в html – елементи th. Для таблиць, що мають два або більше рівня заголовків слід використовувати елементи розмітки, щоб асоціювати заголовки з вмістом, thead, tfoot, tbody для зв'язку рядків, col, colgroup - для зв'язку стовпців, атрибути axis, scope і headers для опису більш складних взаємозв'язків. Використання таблиці, якщо таблиця не має сенсу в лінеаризованому вигляді вважається поганою практикою, або потрібно використовувати альтернативне подання таких таблиць. Всередині таблиць не потрібно використовувати структурні елементи розмітки для цілей візуального форматування. Наприклад, не варто використовувати th для виділення комірки, яка не є заголовком. Резюме таблиці задаються за допомогою атрибуту summary, скорочені заголовки таблиці – атрибут abbr тегу tr. Такі заголовки можуть бути корисні, щоб уникнути повторів довгого заголовка при послідовному зачитування всіх елементів таблиці.

Якщо сторінки використовують нові технології, то потрібно переконатися в їх доступності. Документ має бути прочитаний при відключених стильових сторінках, скриптах апплетах або інших програмних елементах. Обробники подій в скриптах і апплетах не повинні залежати від типу пристрою введення користувача, для цього використовуються події рівня додатку, такі як onblur, onfocus, onselect. Обробники подій, які залежать від пристрою потрібно використовувати парно, onmousedown разом з onkeydown, onmounseup разом з onkeyup, onclick разом з onkeypress. Динамічний вміст має бути доступним в різних клієнтських середовищах, наприклад, серверні скрипти забезпечують більш високу доступність, ніж клієнтські скрипти. Для того, щоб протестувати цей пункт можна використати текстовий браузер lynx, який входить в більшість дистрибутивів Linux, а також має доступні windows

версії. Еквівалент динамічного вмісту повинен оновлюватися при оновленні динамічного контенту. Дана вимога підкреслює, що WAI-ARIA не просто технічний стандарт, за який відповідають розробники сайту. Стандарт в рівній мірі адресований авторам вмісту, редакторам та контент-менеджерам, а положення стандарту повинні бути закріплені в контент-стратегії сайту.

У користувача має бути можливість мати контроль над змінами сторінки, які відбуваються за таймером. Розробникам слід уникати вмісту зі швидким мерехтінням, поки браузери не навчаться його фільтрувати. Людям з фоточутливою формою епілепсії протипоказано дивитися на спалахи з частотою від 4 до 59 Гц. Не варто розміщувати рухомі елементи на сторінці, при необхідності використання руху має бути простий механізм відключення рухомих об’єктів на сторінці. Сторінки з автоматичним періодичним оновленням сторінки можуть дезорієнтувати користувачів асистивних технологій. Поки браузери не навчаться блокувати клієнтську переадресацію, не варто використовувати її автоматично.

Інтерфейс має бути спроектований без прив’язки до конкретного пристрою. Кожен елемент, який має власний інтерфейс повинен бути доступним з різних пристроїв. У скриптах повинні бути задані обробники логічних подій, а не подій, які залежать від типу пристрою введення. Керуючим елементам, посиланням і об’єктам взаємодії повинен бути заданий порядок табуляції за допомогою атрибуту tabindex, якщо елемент не має його за замовчуванням. Важливим посиланням і керуючим кнопкам можна задати гарячі клавіші для покращення користувацького досвіду.

Для сумісності зі старими браузерами допустимі тимчасові рішення. Поки браузери не будуть коректно обробляти порожні елементи введення (input, textarea), дозволяється наповнювати елементи введення заповнювачем. Деякі старі браузери не дозволяють користувачеві потрапити в поле введення, яке не містить тексту. Використання модальних вікон або відкриття посилання в новому вікні без попередження користувача недопустимі.

Асистивні технології повинні мати інформацію про контекст, в якому зараз знаходяться. Великі блоки інформації потрібно розділити на логічні і зрозумілі для сприйняття частини, для цього потрібно використовувати заголовки різного рівня, елементи optgroup, fieldset та інші. Текстові мітки повинні бути ясно асоційовані з відповідними їм елементами, наприклад за допомогою атрибуту for елемента label. Методом переговорів про тип вмісту – content-negotioation, що передбачений протоколом HTTP, можна надати інформацію для того, щоб користувач міг вибрати зручний йому формат вмісту – мова, тип вмісту та інше.

Механізми навігації повинні бути інтуїтивно зрозумілі і ясні. Текст посилання повинен ясно описувати сторінку призначення. У формі мапи сайту або змісту можна підтримувати інформацію про загальну структуру сайту. Пов’язані між собою посилання мають бути згруповані, а користувачу можна надати можливість приховати групу посилань.

Документи мають бути простими і зрозумілими, варто обирати найбільш простий з можливих стилів мови для викладу змісту. Текст можна доповнити ілюстраціями або звуковими файлами для спрощення сприйняття матеріалу. Використання єдиного візуального стилю. для всіх сторінок сайту є загальноприйнятою практикою.

2 ПРОЕКТУВАННЯ СИСТЕМИ

2.1 Постановка технічного завдання

Технічне завдання полягає в дослідженні предметної області, виявленні недоліків існуючих сайтів, які ускладнюють його використання, а також в розробці спеціалізованого інтернет ресурсу у відповідності до ініціативи Консорціуму Всесвітньої павутини щодо забезпечення доступності – WAI

(Web Accessibility Initiative), яка формулює технічні умови, правила, методи і допоміжні ресурси, які описують рішення по забезпеченню веб-доступності. Всі перелічені компоненти є міжнародними стандартами в області вебдоступності.

Для реалізації стандарту було вирішено розробити демонстраційну версію сайту офтальмологічної клініки. Основною цільовою аудиторією сайту будуть люди з тими чи іншими проблемами здоров'я, зокрема значна частка користувачів сайту мають офтальмологічні обмеження, починаючи від астигматизму і закінчуючи повною сліпотою, а отже повний функціонал сайту має бути доступним при використанні асистивних технологій, а також має бути можливість навігації по сайту та його інтерактивним елементам виключно за допомогою клавіатури.

Також потрібно виконати тестування інтерфейсу і проаналізувавши отримані дані надати оцінку ефективності сайту.

2.2 Вимоги до розроблюваної системи

Провівши аналіз завдань, поставлених розроблюваному сайту, я визначив основні вимоги до розроблюваного сайту офтальмологічної клініки:

- представлення клініки «IQ Clinic» в мережі інтернет;
- представлення актуальної інформації про клініку, пропоновані послуги, їх вартість;
- відповідність стандарту доступності WAI-ARIA.

2.3 Вимоги до usability

Відповідно до термінології, usability (юзабіліті) сайту - це ступінь ефективності, продуктивності і зручності взаємодії людини з інтерфейсом.

Також, часто під цим словом об'єднують методи підвищення ефективності роботи web-сайту.

Власник будь-якого сайту бажає бачити на своєму ресурсі якомога більшу кількість відвідувачів. Якщо ж web-сайт розроблений для отримання прибутку, то цифри відвідуваності і конверсії стають життєво необхідними.

Для підвищення якості сайту з точки зору юзабіліті мною був складений перелік критеріїв, якого я дотримувався при розробці клієнтської частини вебсайту медичного закладу. Моїм основним принципом в вирішенні питань юзабіліті стало правило «Чим простіше, тим краще».

Дизайн сайту:

1) простота і мінімалістичність - інформація для користувача представляється в структурованому вигляді і тільки в тому обсязі, який необхідний на даному етапі;

2) доступність - основна інформація і функції сайту (контактна інформація, опис запропонованих послуг, форма заявки повинні бути доступними для користувача на всіх сторінках сайту;

3) актуальність - дизайн сайту повинен відповідати останнім тенденціям в web-дизайні, але в той же час не рясніти графічними ефектами щоб не відволікати користувачів від інформації, що цікавить їх в першу чергу;

4) швидкодія - швидкість завантаження сторінки повинна бути максимальною, але в той же час якість графічних елементів необхідно зберегти на гранично високому рівні;

ґ) зручність навігації. Простота навігації по сайту це один з найважливіших критеріїв при розробці проекту. Користувач повинен інтуїтивно і безперешкодно переміщатися по сайту, отримувати необхідну інформацію або послугу [6];

д) внутрішній зміст сайту. Ресурс повинен містити тільки якісний і повно викладений матеріал з правильно підібраними «Ключовими словами».

2.4 Проектування доступної кольорової палітри

Контраст кольорів - важливий аспект доступності в дизайні продукту. Хороша контрастність полегшує використання продуктів людьми з порушеннями зору і допомагає в недосконалих умовах, таких як слабке освітлення або старі екрани. Досягнення правильного контрасту з кольором є складним завданням і особливо тому, що колір неймовірно суб'єктивний і дуже впливає на естетику продукту.

Існує багато прикладів, коли відвідувачі не можуть чітко розглянути вміст сайту. Вони можуть бути дальтоніками, не здатними відрізнити колір тексту і фону сторінки (1 з 12 чоловіків і 1 з 200 жінок європейського походження). Можливо користувач просто читає текст при яскравому сонячному освітленні, яке створює на екрані безліч відблисків, значно погіршуючи видимість. Або може це просто літня людина з ослабленим зором, не здатним розрізняти кольори так само добре, як раніше [8].

Основна порада щодо доступності кольорів - це вибір кольорів, які можуть розрізнити всі люди. Здатність бачити текст, очевидно, необхідна для його прочитання і, як результат, зрозуміти його зміст. То як ми можемо знати, які кольори відрізняють люди? Вся справа в контрасті, який полягає в різниці між кольором переднього плану та кольором фону. Протилежно контрастні кольори зображені на рисунку 2.1.

Рисунок 2.1 – Контрастні кольори на колірному колесі

У наведеному нижче прикладі текст зліва дуже схожий на колір фону (низький контраст), а текст справа добре відрізняється (високий контраст) і набагато легше читається (рисунок 2.2).

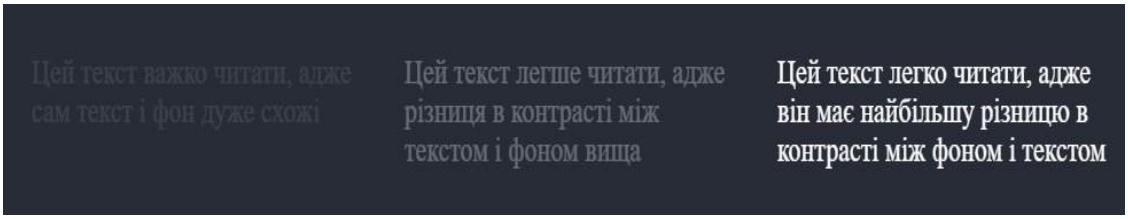

Рисунок 2.2 – Приклад недостатньо контрастного тексту

Тільки 24% сайтів роблять колір всього тексту досить контрастним. Або іншими словами, 3 з 4 сайтів містять текст, який зливається з фоном, що ускладнює його читання.

Рекомендації по доступності веб-контенту, розроблені W3C, представляють формулу для розрахунку контрастності між двома кольорами, з обчисленням коефіцієнта контрастності.

$$(L1 + 0.05) / (L2 + 0.05), \quad (2.1)$$

де L1 – відносна яскравість світліших кольорів, L2 – відносна яскравість темніших кольорів.

Коефіцієнт контрастності варіюється в діапазоні від 1:1 (немає різниці) до 21:1 (максимально можливе значення). Їх легко обчислити в багатьох безкоштовних інструментах, як наприклад Contrast.

WAI-ARIA точно визначає коефіцієнти контрастності для тексту, що читається:

– рівень AA, мінімальний стандарт. У дрібного шрифту коефіцієнт контрастності повинен бути 4.5:1 або вище. У великого шрифту 3:1 або вище;
– рівень AAA, розширений стандарт. У дрібного шрифту коефіцієнт контрастності повинен бути 7:1 або вище. У великого шрифту 4,5:1 або вище.

Як «великий» рахується не напівжирний шрифт розміром 18pt або 24px чи більший, або напівжирний розміром 14pt або 19px чи більший, в іншому випадку це «дрібний» шрифт. W3C визначає точку як 1/72 дюйма, а піксель як 1/96 дюйма, тому для оптимізації зображення в точки помножте значення пікселя на 0,75 [9].

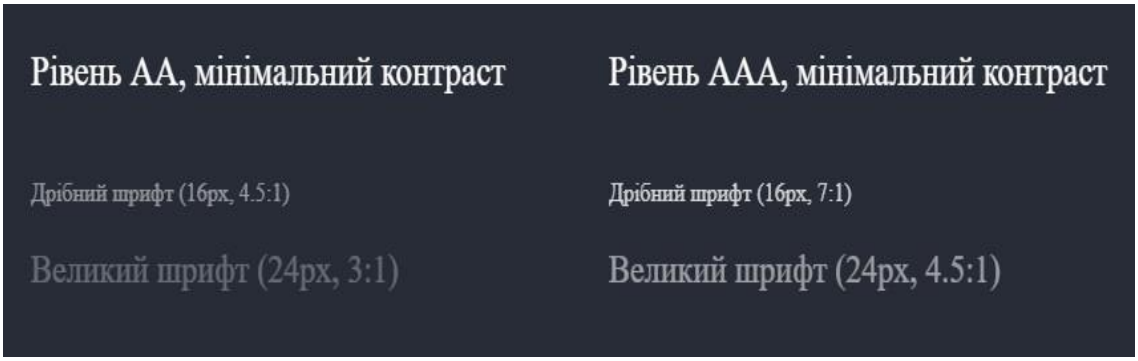

Рисунок 2.3 – Мінімальний контраст тексту за стандартом WAI

Доступний не означає яскравий. Стандарт навмисно фокусує увагу розробників виключно на контрасті між кольором переднього плану і кольором фону, а не на тому наскільки вони є яскравими. Розуміння того, наскільки яскравим виглядає кожен колір, допомагає відрізнити відтінки один від одного. Вибір доступних поєднань кольорів вимагає від кожного окремого дизайнера або інженера розуміння основних принципів вибору пар кольорів з достатньою контрастністю в кожній ситуації [10].

2.5 Зручність читання

2.5.1 Зум і масштабування сторінок

Основна мета веб-сторінки - доставити вміст, з яким хочуть взаємодіяти користувачі. Цей вміст може бути відео або асортиментом зображень, але в більшості випадків це просто текст на сторінці. Надзвичайно важливо, щоб текстовий вміст був розбірливим для читачів. Якщо відвідувачі не можуть прочитати веб-сторінку, вони не можуть працювати з нею, що закінчується виходом зі сторінки.

Використання розбірливого розміру шрифту та достатнього розміру інтерактивних елементів допомагає користувачам читати та взаємодіяти зі сторінкою. Але навіть веб-сайти, які повністю дотримуються всіх цих рекомендацій, не можуть задовольнити конкретні потреби кожного відвідувача. Ось чому такі важливі функції пристрою, як збільшення та збільшення масштабу. Вони дозволяють користувачам адаптувати ваші сторінки, щоб їхні потреби були задоволені. Або у випадку особливо недоступних сайтів, що використовують дрібні шрифти та кнопки, це дає користувачам можливість навіть користуватися сайтом.

Є рідкісні випадки, коли вимкнення масштабування є прийнятним, наприклад, наприклад коли сторінка представляє собою браузерну гру з

сенсорними елементами управління. Якщо в цьому випадку дозволено масштабування, телефони гравців збільшуватимуть і зменшуватимуть сторінку кожного разу, коли гравець двічі натискатиме на гру, зробивши її недоступною.

Через це розробникам надається можливість відключити цю функцію, встановивши одну з наступних двох властивостей у мета тегі viewport:

а) user-scalable надають значення 0 або no;

б) maximum-scale надають значення 1.

На жаль, розробники так зловживали цим, що майже кожний третій сайт на мобільних пристроях відключає цю функцію, і Apple (починаючи з iOS 10) більше не дозволяє веб-розробникам відключати масштабування. Мобільний Safari взагалі ігнорує тег. На нових пристроях iOS можна масштабувати будьякі сторінки за допомогою зуму.

2.5.2 Визначення мови

Мережа наповнена дивовижною кількістю контенту. Однак, у світі існує понад 1000 різних мов, і вміст, який ви шукаєте, може бути написаний не тою, якою ви добре володієте. Останніми роками перекладацькі технології стали значно кращими і, мабуть, кожен користувався хоча б однією з них, наприклад, Google translate.

Щоб полегшити цю функцію, механізми перекладу повинні знати, на якій мові написані ваші сторінки. Це робиться за допомогою атрибута lang. Без цього комп’ютери повинні здогадуватися, якою мовою написана ваша сторінка. Як ви можете уявити, це призводить до багатьох проблем, особливо коли сторінки використовують декілька мов (наприклад, навігація вашою сторінкою є українською мовою, але вміст публікації - англійською мовою).

Ця проблема є ще більш вираженою у таких допоміжних технологіях як синтезатори мовлення, як, наприклад, зчитувачі екрану, де, якщо не вказано жодної мови, вони, як правило, читають текст за типовою мовою користувача.

2.5.3 Відволікаючий вміст

Деякі користувачі, наприклад, люди з когнітивними розладами, мають труднощі в зосередженні на одному і тому ж завданні протягом тривалого періоду часу. Ці користувачі не хочуть мати справу зі сторінками, які містять багато рухомих елементів та анімації, особливо коли ці ефекти є виключно косметичними та не пов'язані із основним призначенням сторінки. Як мінімум, цим користувачам потрібен спосіб вимкнути всі відволікаючі анімації.

На жаль, статистика свідчить про те, що безмежно циклічні анімації є досить поширеними в Інтернеті, і майже кожна п’ята сторінка використовує їх за допомогою нескінченних анімаційних файлів CSS або елементів <marquee> та <blink>.

Однак, слід зауважити що в більшості випадків причиною появи таких проблем є кілька популярних сторонніх бібліотек таблиць стилів, які включають безмежно циклічні анімації CSS за замовчуванням.

2.6 Навігація по сторінці

Коли відвідувачі відкривають веб-сторінку, їх головна мета – знайти інформацію, яка їх найбільше цікавить – причину через яку вони перейши на сторінку в першу чергу. Для того, щоб допомогти користувачам знайти бажаний вміст якнайшвидше і, що важливо, не допустити натискання на кнопку назад, ми намагаємось розділити вміст наших сторінок на кілька візуально різних розділів, шапка сайту для навігації, різні заголовки в статті,

щоб користувачі могли швидко пробігтись по ним очима, футер для іншої додаткової інформації тощо.

Хоча це і надзвичайно важливо, нам потрібно подбати про розмітку наших сторінок, щоб комп'ютери наших відвідувачів могли сприймати і ці окремі розділи. Хоча більшість читачів використовують мишу для навігації по сторінках, багато інших покладаються на клавіатури та зчитувачі екрану. Ці технології значною мірою покладаються на те, наскільки добре їх комп'ютери розуміють розмітку вашої сторінки.

Заголовки одні з найбільш семантично важливих елементів на сторінці. Заголовки корисні не лише візуально, але й для зчитувачів екрану. Вони дозволяють скринрідерам швидко переходити з розділу в розділ і допомагають вказати, де закінчується одна секція і починається інша. Коли людина відвідує сайт за допомогою зчитувача екрану, потрапляє на невідому сторінку, перше, що вона робить на цій сторінці, - це ходить за заголовками. А в екранних зчитувачів є навіть спеціальний режим, який дозволяє зручно це робити. Щоб уникнути плутанини користувачів екранних читалок потрібно переконатися, що ви ніколи не пропускаєте рівень заголовка. Наприклад, не варто перестрибувати від H1 до H3, пропускаючи H2. Тому що це несподівана зміна, яка змусить користувача читача екрану думати, що він пропустив частину вмісту. Це може змусити їх почати шукати все, що вони, можливо, пропустили, навіть якщо це не так. Крім того, дотримання більш послідовного дизайну допоможе всім читачам. Якщо на сторінці є заголовки і відображається чітка структурована ієрархія, то людині буде зручно орієнтуватися на сайті, він зможе швидко перейти до потрібного розділу.

З того часу, як був представлений HTML5, з’явилося чимало html елементів які допомагають браузерам і зчитувачам екрану розуміти розмітку і структуру сторінки. Правильне використання смислових тегів із п’ятого стандарту HTML гарантує, що ваш верстка буде логічно інтерпретована

більшістю браузерів та допоміжних технологій.

Такі елементи, як <header>, <footer>, <nav> і <main>, вказують, у якій частині знаходиться певний тип вмісту, і дозволяють користувачам швидко переходити по сторінці. Вони широко використовуються в розробці, більшість з яких використовується на більш ніж 50% сторінок (тег <main> - виняток).

Інші теги, такі як <article>, <hr> та <aside>, допомагають користувачам зрозуміти основний вміст сторінки. Наприклад, вони повідомляють, де закінчується одна стаття й починається інша. Ці елменти використовуються не так часто. Кожен має показник близько 20%. Вони не є невід'ємною частиною сторінки.

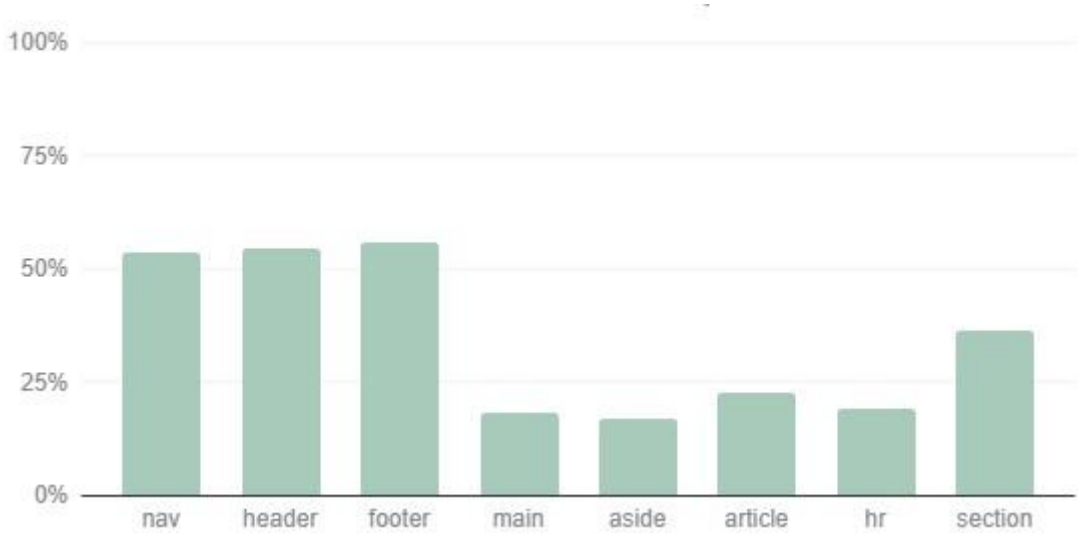

Рисунок 2.4 □ Частота використання смислових елементів

Усі ці елементи розроблені насамперед для поліпшення доступності та не виділяються візуально, а це означає, що ви можете замінити їх на використовувані в даний час елементи, не побоюючись непередбачених наслідків.

Багато популярних читачів екранів дозволяють користувачам швидко переходити через посилання, списки, елементи списку, фрейми та елементи форми, такі як кнопки, радіо кнопки, чекбокси.

Таблиці - один з основних способів організації та подання великої кількості даних. Багато допоміжних технологій, такі як зчитувачі екранів і перемикачі (світчі), які можуть бути корисними користувачам з порушеннями опорно-рухового апарату, можуть мати спеціальні функції, які дозволяють їм ефективніше переміщатися по табличним даним.

Залежно від конкретної структури певної таблиці, використання заголовків таблиць полегшує читання через стовпці та рядки, не втрачаючи контексту того, до яких даних відноситься цей конкретний стовпець чи рядок. Для користувачів читалок переміщення по таблиці без заголовків рядків і стовпців є доволі непростою задачею. Це тому, що читачеві екрану важко відслідковувати відповідність комірки до заголовків, особливо коли таблиця досить велика. Щоб розмітити табличні заголовки потрібно просто використати тег <th> замість <td>, або ARIA-ролі columnheader чи rowheader.

Підписи таблиць із використанням елемента <caption> корисні для того, щоб забезпечити більш широкий описовий контекст для читачів усіх типів. Підпис може підготувати читача до сприйняття інформації, яку містить таблиця, і це може бути особливо корисно для людей, яких легко відволікати чи переривати. Вони також корисні людям, які можуть загубитися всередині великої таблиці, наприклад, користувачі, які зчитувачів екрану, або люди з обмеженнями здатності до навчання. Чим простіше читачам зрозуміти, які дані їм подаються, тим краще [11].

2.7 Сумісність з асистивними технологіями, використання ARIA

Однією з найпопулярніших і широко використовуваних специфікацій щодо доступності в Інтернеті є стандарт Accessible Rich Internet Applications

(ARIA). Цей стандарт пропонує великий набір додаткових атрибутів HTML, які допомагають передати роль візуальних елементів (тобто їх семантичне значення) та на які дії вони здатні.

Правильне та належне використання ARIA може бути складним завданням. Досить часто розробники, які почали вивчати тему веб доступності використовують неправильні значення для aria атрибутів. Ця проблема актуальна, оскільки будь-яка помилка у використанні атрибуту ARIA не впливає візуально на сторінку. Деякі з цих помилок можна виявити за допомогою автоматизованого інструменту перевірки, але зазвичай вони потребують практичного використання реального допоміжного програмного забезпечення, як наприклад, зчитувач екрану.

Атрибут "role" є найважливішим атрибутом у всій специфікації ARIA. Він використовується для інформування браузера про призначення даного елемента HTML, тобто про його семантичне значення. Наприклад, елементу , візуально оформленому як посилання за допомогою CSS, слід надати ARIA роль «link».

У стандарті є багато різних ролей, і вони поділяються на різні групи. Наприклад, роль віджетів. Вони призначаються елементам, які є самостійними частинами інтерфейсу користувача. Це звичайні кнопки, радіо кнопки, таби, тултипи. Далі йдуть збірні ролі , які об'єднують елементи з інших ролей. Очевидно, що radiogroup складається з елементів radio, a tablist складається з tab. Наступними йдуть орієнтири і структурні ролі. Ці ролі надаються великим смисловим розділам на сторінці

Окрім ролей, в стандарті ARIA також передбачені стани. Вони можуть описати, який статус має елемент на даний момент. Наприклад, стан для віджетів, вони вказують, що елемент прихований, позначений, заблокований тощо. Існують також стани, які вказують на зв’язок елементів. Наприклад,

елемент описує інший елемент або керує іншим елементом. Також стани можуть декларувати особливі області, в яких очікується зміна вмісту без перезавантаження сторінки. Ці області називаються live regions.

Ролей і станів багато, повний їх опис можна прочитати на сайті специфікації. Найбільш часто використовувані атрибути будуть розглянуті в розділі реалізації системи.

3 РЕАЛІЗАЦІЯ СИСТЕМИ НА ОСНОВІ ТЕХНОЛОГІЧНОГО СТАНДАРТУ «WAI-ARIA»

3.1 Програмні компоненти

В лістингу 3.1 представлено розмітка багаторівневого меню навігації.

Окрім різних ролей, серед яких є структурні ролі navigation та menubar в меню навігації є складові ролі вкладені один в одного. Є декілька станів, наприклад, aria-haspopup зі значенням true говорить про те, що цей елемент має випадаюче меню, а aria-live надає інформацію скринрідеру, що цей елемент динамічний і має властивість змінюватися без перезавантаження сторінки, тепер скринрідер буде відстежувати зміни в цьому блоці і якщо відбулася зміна, то він надасть характерний сигнал, який свідчить про зміну динамічного вмісту і зачитає користувачеві новий вміст блоку. Якщо в цей час скринрідер читав вміст на сторінці, то читання призупиниться і почнеться читання в live-region. Атрибут aria-current використовується для імітації
«хлібних крихт» щоб показати де знаходиться користувач відносно інших сторінок сайту.

Лістинг 3.1

```
<nav class="main-navigation" role="navigation">
 <ul class="nav-list" role="menubar">
  <li class="services" role="menuitem">
   <button type="button" aria-haspopup="true">Услуги</button>
   <ul class="menu-drop" role="menu">
    <li><a href="diagnostics.html">Диагностика и консультация</a></li>
    <li aria-current="page"><a href="laser.html">Лазерная коррекция зрения</a></li>
    <li><a href="uncontact-laser.html">Бесконтактная лазерная коррекция зрения</a></li>
    <li><a href="cataracta.html">Лечение катаракты</a></li>
    <li><a href="glaucoma.html">Лечение глаукомы</a></li>
    <li><a href="retina.html">Лазерное лечение заболеваний сетчатки</a></li>
    <li><a href="surgery.html">Рефракционная хирургия</a></li>
    <li><a href="children.html">Детская офтальмология</a></li>
    <li><a href="optics.html">Оптика</a></li>
  </ul>
  </li>
  <li role="menuitem"><a href="prices.html">Цены</a></li>
  <li role="menuitem"><a href="equipment.html">Оборудование</a></li>
  <li role="menuitem"><a href="doctors.html">Врачи</a></li>
  <li role="menuitem"><a href="reviews.html">Отзывы</a></li>
  <li role="menuitem"><a href="news.html">Новости</a></li>
  <li role="menuitem"><a href="contacts.html">Контакты</a></li>
  <li role="menuitem"><a href="stocks.html">Акции</a></li>
  <li role="menuitem"><a class="login" href="login.html">Вход</a></li>
 </ul>
</nav>
```

В лістингу 3.2 представлено код форми пошуку. Атрибут placeholder є недоступним для програм читання з екрану, тому підказку про призначення поля вводу слід продублювати в атрибуті aria-label.

Лістинг 3.2

```
<form action="" method="get">
```

```
<input class="search-field" type="search" aria-label="Поиск информации" placeholder="Поиск...">
<button class="search-button" type="submit" aria-label="Кнопка поиска"></button> </form>
```

Кнопка пошуку візуально замінена на декоративну іконку, яка має приємний вигляд з точки зору дизайну для звичайних користувачів, але зовсім неінформативна для асистивних технологій. Тому потрібно приховувати декоративні елементи, які не несуть в собі ніякої інформації від асистивних технологій, аби не заплутувати користувача і надати доступ до атрибуту arialabel з текстовим описанням кнопки. Для цього використовують CSS сніппет visually-hidden, що представлений в лістингу 3.3.

Лістинг 3.3

```
.visually-hidden {
position: absolute;
width: 1px;
height: 1px;
margin: -1px;
border: 0;
padding: 0;     clip:
rect(0 0 0 0);
overflow: hidden;
}
```

В лістингу 3.4 представлена розмітка навігаційних елементів таб меню.

Лістинг 3.4

```
<div class="tab" role="tablist">
        <button class="tablinks" onclick="openTab(event, 'about')" id="defaultOpen">Наш центр</button>
        <button class="tablinks" onclick="openTab(event, 'equipment')">Оборудование</button>
        <button class="tablinks" onclick="openTab(event, 'reviews')">Отзывы</button>  </div>
```

Javascript код для взаємодії між кнопками табу та його секціями, а також для реалізації можливості повного використання функціоналу таб меню за допомогою однієї клавіатури, без використання миші представлено в лістингу 3.5.

Лістинг 3.5

```
let dropMenuButton = document.querySelector(".services > button");
let dropMenu = document.querySelector(".menu-drop");
dropMenuButton.addEventListener("click", function(evt) {
evt.preventDefault();
  dropMenu.classList.toggle("active");
});
function openTab(evt, tabName) {
  // Declare all variables
  var i, tabcontent, tablinks;
  // Get all elements with class="tabcontent" and hide them
tabcontent = document.getElementsByClassName("tabcontent");
for (i = 0; i < tabcontent.length; i++) {  tabcontent[i].style.display
= "none";
  }
  // Get all elements with class="tablinks" and remove the class "active"
tablinks = document.getElementsByClassName("tablinks");  for (i = 0;
i < tablinks.length; i++) {  tablinks[i].className =
tablinks[i].className.replace(" active", "");
  }
  // Show the current tab, and add an "active" class to the link that opened the tab
document.getElementById(tabName).style.display = "block";
evt.currentTarget.className += " active";
} document.getElementById("defaultOpen").click();
```

Коли користувач взаємодіє з модальним вікном і навігується між елементами за допомогою клавіатури, важливо подбати про те, щоб навігаційний фокус не виходив за межі модального вікна, допоки користувач сам не закриє його, в іншому випадку незрячі користувачі будуть дезорієнтовані неочікуваною поведінкою фокуса і не зрозуміють як саме вони

опинилися за межами вікна з яким вони працювали. Для впровадження цього функціоналу розробляється атрибут inert, який скоріш за все буде імплементований в наступній специфікації HTML, на даний момент атрибут не підтримується більшістю браузерів, а тому доводиться власноруч створювати javascript поліфіл, код якого представлено в лістингу 3.6.

Лістинг 3.6

```
ensureUntabbable() {
  if (this.node.nodeType !== Node.ELEMENT_NODE) {
return;
  }
  const element = /** @type {!Element} */ (this.node);
if (matches.call(element, _focusableElementsString)) {
if (/** @type {!HTMLElement} */ (element).tabIndex
=== -1 && this.hasSavedTabIndex) { return;
    }
    if (element.hasAttribute('tabindex')) { this._savedTabIndex = /** @type
{!HTMLElement} */ (element).tabIndex;
    }
    element.setAttribute('tabindex', '-1');
    if (element.nodeType === Node.ELEMENT_NODE) {
element.focus = function() {};
this._overrodeFocusMethod = true;
    }
  } else if (element.hasAttribute('tabindex')) { this._savedTabIndex = /**
@type {!HTMLElement} */ (element).tabIndex;
element.removeAttribute('tabindex');
  }
 }
if (!Element.prototype.hasOwnProperty('inert')) {
Object.defineProperty(Element.prototype, 'inert', {
enumerable: true, /** @this {!Element} */ get:
function() { return this.hasAttribute('inert');
  },
  /** @this {!Element} */ set:
function(inert) {
inertManager.setInert(this, inert);
  },
```

```
  });                    }                    function
dispatchTabEvent(opt_shiftKey) {     var ev
= null;     try {
       ev = new KeyboardEvent('keydown', {
keyCode: 9,              which: 9,              key:
'Tab',          code: 'Tab',          keyIdentifier:
'U+0009',             bubbles: true
       });
    } catch (e) {
       try {
          //Internet Explorer             ev =
document.createEvent('KeyboardEvent');
ev.initKeyboardEvent(
               'keydown',
true,                 true,
window,
               'Tab',                0,
opt_shiftKey ? 'Shift' : '',
false,
               'en'
               )
       } catch (e) {}
    }        if
(ev) {
       Object.defineProperty(ev, 'keyCode', { value: 9});
    }                  catch                  (e)                  {}
document.dispatchTabEvent(ev);
}
```

В лістингу 3.7 представлено код реалізації відкриття і закриття з клавіатури спливаючих вікон за допомогою обробника подій для елементів з вбудованими подіями onclick, таких як кнопки або посилання.

Лістинг 3.7

```
let popupButton = document.querySelector(".booking"); let
popupForm = document.querySelector(".booking-popup"); let
closePopup = document.querySelector(".close-popup"); let
```

```
nameInput = document.querySelector(".name-input");
popupButton.addEventListener("click", function(evt) {
evt.preventDefault();
   popupForm.classList.add("active");
nameInput.focus();
}); closePopup.addEventListener("click",
function(evt) {     evt.preventDefault();
popupForm.classList.remove("active");

});
```

Так само як в звичаній розмітці зв’язуються поля форм з підписами до форм за допомогою атрибутів for для тегу label і id для тегу input. Аналогічно поля форми мають бути пов’язані з підписами для скрінридера, щоб в момент фокусування на полі форми незрячі користувачі могли почути який тип даних потрібно ввести. Приклад використання aria атрибутів для полів форм представлений в лістингу 3.8.

Лістинг 3.8

```
<div class="booking-popup">
        <p>Записаться на приём</p>
        <form class="popup-form" action="" method="post">
          <label class="name-label" for="name-field">Ваше имя (обязательно)</label>
          <input class="name-input" id="name-field" type="text">
          <label for="email-field">Ваш E-mail (обязательно)</label>
          <input id="email-field" type="email" required>
          <label for="phone-field">Ваш телефон (обязательно)</label>
          <input id="phone-field" type="tel" required>
          <label for="reason-field">Причина обращения</label>
          <textarea name="reason" id="reason-field" required></textarea>
          <div class="form-buttons">
            <button type="button" class="close-popup">Закрыть окно</button>
            <button class="send-popup">Записаться</button>
          </div>
        </form>
      </div>
```

3.2 Тестування системи

В браузері Google Chrome є вбудований аудит, який дозволяє оцінити рівень доступності, а також надає поради по поліпшенню безбар’єрного доступу до ресурсу. В цілому тест перевіряє на:

– валідність розмітки для дотримання доступності для користувачів програм зчитування з екрану;

– дотримання достатнього коефіцієнту контрастності на сторінці, розміру шрифтів;

– унікальність id атрибутів на сторінці;

– наявність мета-тегів lang, title, meta;

– відсутність заблокованого зуму;

– наявність та коректність використання смислових елементів, таких як header, main, footer, nav, section, article, aside та ін.

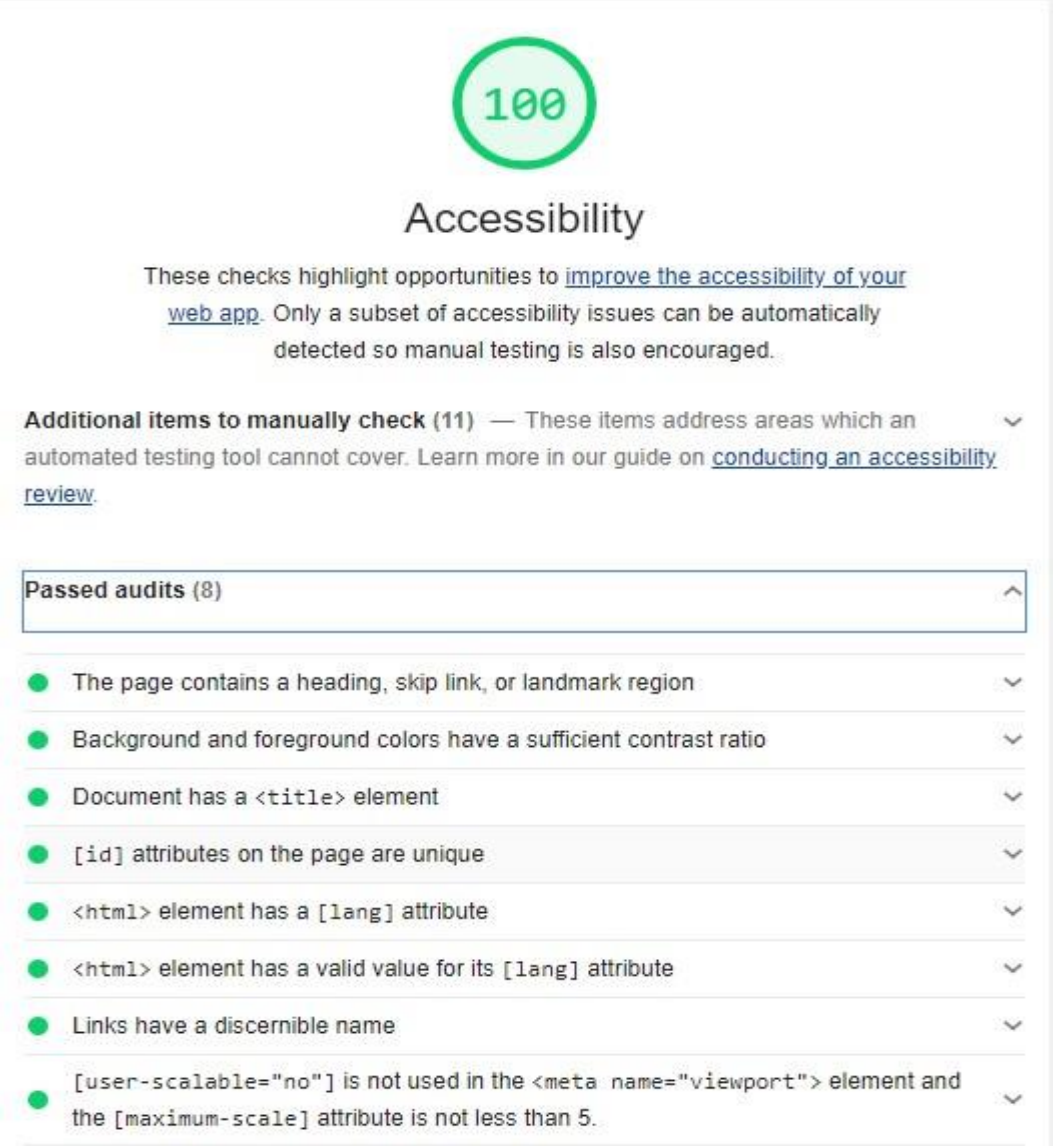

Рисунок 3.9 Результат аудиту на доступність в Chrome dev tools

ВИСНОВКИ

У ході виконання дослідницької частини роботи була розглянута проблематика та актуальність доступних веб-інтерфейсів в мережі, їхній вплив на соціально-економічні і культурні права людей, які їх потребують.

Були розглянуті правові і фінансові чинники впровадження інклюзивних систем в різних країнах та їх взаємозв’язок між собою. Було виявлено, що такі системи мають більш широке охоплення користувачів і, як наслідок, кращі

фінансові показники, в тому числі завдяки уникненню отримання штрафів з боку контролюючих органів в країнах з жорстким законодавством в області захисту прав людини.

Був проведений огляд допоміжних і апаратних технологій та аналіз основних принципів їх взаємодії з методами і практиками, описаними в технологічному стандарті «WAI-ARIA», який визначає підходи до того, щоб зробити вміст сайтів та інтернет-додатків більш доступними для людей з обмеженими можливостями.

На основі проведеного аналізу було вирішено розробити спеціалізований веб-сайт офтальмологічної клініки через високу частку користувачів, які будуть зацікавлені в адаптації інтерфейсу та вмісту сайту до їх фізичних потреб.

На етапі проектування були розглянуті основні принципи розробки доступної кольорової палітри, зручності читання вмісту, навігації по сторінці і сумісності з асистивними технологіями.

В ході роботи було реалізовано веб-сайт офтальмологічної клініки, який відповідає вимогам технологічного стандарту «WAI-ARIA» рівня AAA.

Переваги розробленої системи:

1) коректна робота всіх функцій сайту з асистивними технологіями, що надає можливість користування всіма функціями сайту незрячим людям;

2) можливість повної взаємодії з сайтом виключно за допомогою клавіатури, що надає можливість взаємодії з сайтом користувачам з проблемами опорно-рухового апарату;

3) доступна кольорова палітра елементів сайту для людей з колірною сліпотою;

4) нижча собівартість підтримки сайту в майбутньому за рахунок використання валідної і семантичної верстки;

5) відповідність до законодавства низки країн в області веб-доступності.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Diggs J. Accessible Rich Internet Applications (WAI-ARIA) 1.1 [Електронний ресурс] / J. Diggs, S. McCarron, M. Cooper. – 2017. – Режим доступу до ресурсу: https://www.w3.org/TR/wai-aria-1.1/.

2. Horton S. A Web for Everyone: Designing Accessible User Experiences / S. Horton, W. Quesenbery., 2014. – 288 с.

3. Pickering H. Inclusive Design Patterns: Coding Accessibility Into Web Design / Heydon Pickering., 2016. – 312 с. – ISBN 3945749433.

4. Joshue O. Pro HTML5 Accessibility: Building an Inclusive Web / O. Connor Joshue., 2012. – 386 с. – ISBN 1430241942.

5. Accessibility [Електронний ресурс] / M.Kearney, D. Gash, A. Boxhall, R. Dodson – Режим доступу до ресурсу: https://developers.google.com/web/fundamentals/accessibility.

6. Horton S. Access by Design: A Guide to Universal Usability for Web Designers / Sarah Horton., 2005. – 264 с. – ISBN 032131140X.

7. Web Accessibility: Web Standards and Regulatory Compliance / J.Thatcher, C. Heilmann, R. Rutter, B. Lawson., 2006. – 648 с. – ISBN 1590596382.

8. Батогов Д. Доступность интерфейсов.[Електронний ресурс] / Дмитрий Батогов. – 2018. – Режим доступу до ресурсу: https://habr.com/ru/company/yandex/blog/424879/.

9. Geley L. UX and the Importance of Web Accessibility [Електронний ресурс] / Laura Geley. – 2017. – Режим доступу до ресурсу: https://www.toptal.com/designers/ui/importance-web-accessibility.

10. Koopersmith D. Designing accessible color systems [Електронний ресурс] / D. Koopersmith, W. Miner. – 2019. – Режим доступу до ресурсу: https://stripe.com/blog/accessible-color-systems.

11. Accessibility [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: https://developer.mozilla.org/uk/docs/Web/Accessibility.

12. Paisios N. Accessibility [Електронний ресурс] / N. Paisios, D. Fox,

A. Klein. – 2019. – Режим доступу до ресурсу: https://almanac.httparchive.org/en/2019/accessibility.

13. Holmes K. Mismatch: How Inclusion Shapes Design / K. Holmes, J. Maeda., 2018. – 184 с. – ISBN 0262038889.