

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)

Кафедра Інформатики  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти другий (магістерський)

**ДОСЛІДЖЕННЯ МЕТОДІВ АДАПТАЦІЇ ВЕБЗАСТОСУНКІВ**  
**ДЛЯ КОРИСТУВАЧІВ З ОБМЕЖЕНИМИ МОЖЛИВОСТЯМИ**  
(тема)

Виконав:  
здобувач 2 року навчання,  
групи ІНФМ-24-1  
Денисенко А. В.  
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки  
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика  
(повна назва освітньої програми)

Науковий керівник доц. Тітова О. В.  
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики \_\_\_\_\_  
(підпис)

Кобилін О. А.  
(прізвище, ініціали)

2025 р.

## Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджментуКафедра ІнформатикиРівень вищої освіти другий (магістерський)Спеціальність 122 Комп'ютерні науки  
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

«\_\_\_\_\_» \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУздобувачеві Денисенку Артему Віталійовичу  
(прізвище, ім'я, по батькові)1. Тема роботи Дослідження методів адаптації вебзастосунків для користувачів з обмеженими можливостями

затверджена наказом університету від 14 листопада 2025 року № 1045Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 14 грудня 2025 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, засоби для програмної реалізації React, редактор коду Visual Studio Code.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1. Аналіз сучасних методів адаптації вебзастосунків.

2. Аналіз літературних джерел щодо апробації методів адаптації вебзастосунків під потреби користувачів.

3. Формування покрокового алгоритму для кожного із вибраних методів адаптації.

4. Візуалізація сформованих покрокових алгоритмів.

5. Розробка вебзастосунку, що надасть змогу дослідити методи адаптації вебзастосунків для користувачів з обмеженими можливостями.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми адаптації вебзастосунків, об'єкт та мета дослідження, постановка задачі, блок-схема алгоритму адаптації вебзастосунків, ілюстрація головного екрану розробленого вебзастосунку, висновки, перспективи та апробація роботи.

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

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	29.09.2025	
2	Аналіз завдання, підбір літератури	30.09.25-07.10.25	
3	Аналіз літератури з досліджуваної проблеми	08.10.25-14.10.25	
4	Особливості методів адаптації вебзастосунків для користувачів з обмеженими можливостями	15.10.25-20.10.25	
5	Дослідження методів адаптації вебзастосунків для користувачів з обмеженими можливостями	21.10.25-27.10.25	
6	Програмна реалізація	28.10.25-29.12.25	
7	Обґрунтування отриманих результатів	30.11.25-05.12.25	
8	Оформлення пояснювальної записки	06.12.25-10.12.25	
9	Перевірка на нормоконтроль	10.12.25-14.12.25	
10	Перевірка на плагіат	10.12.25-14.12.25	
11	Рецензування	10.12.25-14.12.25	
12	Підготовка презентації та доповіді	10.12.25-22.12.25	
13	Занесення роботи в електронний архів	10.12.25-22.12.25	
14	Попередній захист кваліфікаційної роботи	23.12.25	

Дата видачі завдання 29 вересня 2025 р.

Здобувач \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

доц. Тітова О. В.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 68 с., 3 рис., 47 джерела.

АДАПТИВНИЙ ІНТЕРФЕЙС, АСИСТИВНІ ТЕХНОЛОГІЇ, ВЕБДИЗАЙН, ДОСТУПНІСТЬ ВЕБЗАСТОСУНКІВ, КОРИСТУВАЧІ З ОБМЕЖЕНИМИ МОЖЛИВОСТЯМИ, МОДИФІКАЦІЯ ІНТЕРФЕЙСУ, ПЕРСОНАЛІЗАЦІЯ, СЕМАНТИЧНА РОЗМІТКА, UX-ДОСТУПНІСТЬ.

Об'єктом дослідження є процес адаптації вебзастосунків до потреб користувачів з обмеженими можливостями.

Метою дослідження є розроблення та обґрунтування методу адаптації вебзастосунку для користувачів з обмеженими можливостями, який забезпечить підвищення доступності та якості взаємодії.

Використано методи системного та структурного аналізу, моделювання. У роботі розроблено програмну реалізацію методу адаптації, що передбачає формування профілів користувачів, модифікацію візуальних і навігаційних параметрів інтерфейсу.

Наукова новизна роботи полягає у створенні методу адаптації вебінтерфейсу, що поєднує персоналізований підхід формування доступності та інтеграцію механізмів зміни інтерфейсу з урахуванням типу порушення.

Взаємозв'язок з іншими роботами полягає в узгодженні з сучасними дослідженнями у сфері вебдоступності та доповнює їх підходом динамічної адаптації інтерфейсу на основі профілю користувача.

Рекомендації щодо використання результатів роботи сформовано і подано у висновках.

У результаті дослідження розроблений вебінтерфейс, що автоматично змінює параметри доступності та підвищує відповідність вимогам WCAG 2.1.

## ABSTRACT

Explanatory note to the qualification work: 68 pages, 3 figures, 47 sources.

ADAPTIVE INTERFACE, ASSISTIVE TECHNOLOGIES, WEB APPLICATION ACCESSIBILITY, WEB DESIGN, INTERFACE MODIFICATION, PERSONALIZATION, SEMANTIC MARKUP, USERS WITH DISABILITIES, UX ACCESSIBILITY.

The object of the research is the process of adapting web applications to the needs of users with disabilities.

The aim of the research is to develop and justify a method for adapting web applications for users with disabilities, which will improve accessibility and quality of interaction.

Methods of systematic and structural analysis and modeling were used. The paper develops a software implementation of the adaptation method, which involves the creation of user profiles and the modification of visual and navigation parameters of the interface.

Scientific novelty of the work lies in the creation of a method for adapting the web interface, combining a personalized approach to accessibility and the integration of interface change mechanisms, taking into account the type of impairment.

Interconnection with other works lies in its alignment with current research in the field of web accessibility and complements them with an approach of dynamic interface adaptation based on user profiles.

Recommendations for using the results of the work have been formulated and presented in the conclusions.

As a result of the research, an adaptive web interface was developed that automatically changes accessibility settings and improves compliance with WCAG 2.1 requirements

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	8
Вступ.....	9
<b>1 Огляд досліджуваної проблеми та теоретичні основи адаптації вебзастосунків для користувачів з обмеженими можливостями.....</b>	<b>11</b>
1.1 Особливості взаємодії користувачів з обмеженими можливостями з вебзастосунками .....	11
1.2 Види бар'єрів доступності та їх класифікація .....	13
1.3 Міжнародні стандарти доступності.....	16
1.4 Технології асистивного доступу.....	18
1.5 Методи забезпечення доступності в сучасних вебзастосунках .....	21
1.6 Аналіз існуючих підходів, досліджень і програмних рішень .....	23
1.7 Постановка задачі дослідження.....	26
<b>2 Методи, моделі та технологічні підходи до адаптації вебзастосунків.....</b>	<b>28</b>
2.1 Формалізація вимог до адаптивності і доступних вебінтерфейсів ..	28
2.2 Методи модифікації фронтенду під потреби користувачів з обмеженими можливостями .....	30
2.2.1 Адаптація візуальної складової.....	30
2.2.2 Адаптація навігації та структури інтерфейсу.....	32
2.2.3 Адаптація контенту та інтерактивних елементів .....	33
2.3 Підходи до автоматичного виявлення проблем доступності.....	34
2.4 Моделі персоналізації вебінтерфейсів на основі профілю користувача.....	36
2.5 Використання машинного навчання для покращення адаптації інтерфейсів.....	38
2.6 Аналіз інструментів розроблення доступних вебзастосунків.....	41
2.7 Розроблення методики оцінювання доступності вебзастосунків ....	43

3 Розроблення та дослідження методу адаптації вебзастосунку для користувачів з обмеженими можливостями.....	46
3.1 Обґрунтування вибору середовища програмної реалізації.....	46
3.2 Програмна реалізація .....	49
3.3 Інструкція користувача .....	53
3.4 Дослідження та тестування розробленої моделі.....	57
Висновки.....	60
Перелік джерел посилання .....	62

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

Скрінрідер - програма, що озвучує або виводить шрифтом Брайля інформацію зі сторінки для людей з порушенням зору

ARIA – Accessible Rich Internet Applications (набір атрибутів для забезпечення доступності складних вебінтерфейсів та їхньої взаємодії зі скрінрідерами)

CI/CD – Continuous Integration / Continuous Deployment (безперервна інтеграція та деплой, згадується у контексті тестування доступності)

CSS3 – Cascading Style Sheets version 3 (технологія для стилізації вебсторінок)

DOM – Document Object Model (модель представлення структури вебсторінки у вигляді дерева вузлів)

HTML5 – HyperText Markup Language version 5 (мова гіпертекстової розмітки, що використовується для структурування вебсторінок)

IDE – Integrated Development Environment (інтегроване середовище розробки)

JavaScript – мова програмування, що забезпечує динамічну поведінку вебінтерфейсів

Lighthouse – інструмент автоматичного аналізу якості вебзастосунку, включно з доступністю

React – бібліотека для створення компонентно-орієнтованих вебінтерфейсів

WCAG – Web Content Accessibility Guidelines (міжнародні рекомендації щодо доступності вебконтенту)

## ВСТУП

У сучасному цифровому суспільстві вебзастосунки стали ключовим інструментом комунікації, доступу до інформації, отримання освітніх та адміністративних послуг, здійснення фінансових операцій та інтерактивної взаємодії з різноманітними сервісами. Зростання ролі цифрових платформ супроводжується необхідністю забезпечення їх доступності для всіх категорій користувачів, зокрема для осіб з обмеженими можливостями. За оцінками Всесвітньої організації охорони здоров'я, понад мільярд людей у світі мають ті чи інші форми інвалідності, що безпосередньо впливає на їх можливість ефективно користуватися вебресурсами. В умовах цифрової трансформації державних сервісів, розвитку електронної комерції, дистанційної освіти та онлайн-комунікацій питання доступності вебзастосунків набуває не лише соціальної, але й економічної та правової значущості.

Сучасний стан досліджень у сфері вебдоступності характеризується активним розвитком міжнародних стандартів та рекомендацій, таких як Web Content Accessibility Guidelines (WCAG), Section 508, EN 301 549. На рівні інструментів розробки поширеними є бібліотеки та сервіси автоматичного аналізу доступності (axe, WAVE, Lighthouse), а також техніки формування семантично коректної структури HTML-документів, ARIA-атрибутів та адаптивних стилів. Попри наявність усталених рекомендацій, дослідники все частіше звертають увагу на недостатність універсальних підходів, які б дозволяли безпосередньо адаптувати інтерфейси під індивідуальні потреби користувачів. Нині активно розвиваються технології персоналізації, асистивні інтерфейси, адаптивні UI-компоненти, а також методи машинного навчання, здатні автоматично виявляти проблеми доступності або пропонувати способи їх усунення.

Водночас аналіз сучасних програмних рішень свідчить про наявність низки проблем, що залишаються відкритими. Більшість інструментів

забезпечують лише перевірку відповідності вебзастосунку окремим критеріям WCAG, але не надають механізмів безпосередньої адаптації інтерфейсу під індивідуальні можливості конкретного користувача. Часто такі рішення не враховують контекст використання, тип порушення, рівень цифрових навичок або доступність на різних пристроях. Значна частина доступних інструментів працює на рівні статичного аналізу і не реагує на зміну поведінки користувача або його взаємодії з інтерфейсом у реальному часі. Це формує потребу у методах, здатних динамічно адаптувати компоненти вебінтерфейсу та забезпечити персоналізовану взаємодію.

Задача, що розглядається в даній роботі, полягає у дослідженні, розробленні та експериментальній оцінці методу адаптації вебзастосунку для користувачів з обмеженими можливостями, який дозволить підвищити доступність та якість взаємодії за рахунок застосування сучасних технологічних та методичних підходів. Для її розв'язання необхідно систематизувати існуючі бар'єри доступності, проаналізувати теоретичні та практичні основи адаптації вебінтерфейсів, визначити недоліки існуючих рішень та сформулювати власний метод адаптації, заснований на поєднанні механізмів персоналізації й інструментів забезпечення доступності.

Актуальність роботи полягає у необхідності усунення бар'єрів, що перешкоджають взаємодії користувачів з порушеннями зору, слуху, моторики або когнітивних функцій із сучасними вебінтерфейсами. Багато існуючих вебзастосунків створюються без урахування специфічних потреб таких користувачів, що призводить до ускладненої навігації, недоступності елементів управління, відсутності коректної семантики або адаптивних механізмів. Нерідко проблеми доступності виявляються лише на етапі тестування або експлуатації, що ускладнює їх усунення та підвищує витрати на доопрацювання. У межах євроінтеграційного курсу України, гармонізації стандартів цифрової інклюзивності та реалізації принципів «доступності за замовчуванням» тема адаптації вебзастосунків стає особливо важливою.

# 1 ОГЛЯД ДОСЛІДЖУВАЛЬНОЇ ПРОБЛЕМИ ТА ТЕОРЕТИЧНІ ОСНОВИ АДАПТАЦІЇ ВЕБЗАСТОСУНКІВ ДЛЯ КОРИСТУВАЧІВ З ОБМЕЖЕНИМИ МОЖЛИВОСТЯМИ

## 1.1 Особливості взаємодії користувачів з обмеженими можливостями з вебзастосунками

Взаємодія користувачів з обмеженими можливостями з вебзастосунками має низку специфічних особливостей, пов'язаних із обмеженнями сенсорних, моторних або когнітивних функцій. Ці особливості визначають спосіб, у який користувач отримує, обробляє та інтерпретує інформацію, а також виконання ним дій у межах вебінтерфейсу. У контексті сучасних цифрових платформ традиційні підходи до проєктування вебінтерфейсів – орієнтація на середньостатистичного користувача, залежність від візуальних елементів, використання дрібних інтерактивних компонентів чи складних жестових патернів – часто виявляються недостатніми або неефективними [1].

Однією з найважливіших груп користувачів є люди з порушеннями зору. Вони можуть мати часткову або повну втрату зорової функції, що потребує застосування скрінрідерів, збільшення масштабу, підвищеного контрасту або альтернативних текстових описів. У таких випадках ключовим чинником стає семантична структура документа, правильне використання ARIA-атрибутів, коректна побудова заголовків, списків та взаємодіючих елементів. Невірно розмічена сторінка або елементи інтерфейсу, що не мають текстових альтернатив, ускладнюють або роблять неможливою навігацію. Крім того, користувачі зі зниженим зором часто користуються клавіатурною навігацією, а тому потребують чіткої логіки фокусування, зрозумілої структури табуляції та можливості керування всіма елементами без миші.

Користувачі з порушеннями слуху взаємодіють із вебзастосунками передусім візуально, однак стикаються з проблемами під час роботи з

мультимедійними матеріалами. Відсутність субтитрів, транскриптів або дублювання аудіоінформації у текстовому вигляді різко знижує доступність таких компонентів. Окрім того, важливо забезпечити доступні повідомлення про події, попередження та помилки, які у типових інтерфейсах часто подаються аудіосигналами. Правильне дублювання таких повідомлень у текстовій формі або у вигляді візуальних маркерів дозволяє користувачам з порушеннями слуху повноцінно взаємодіяти із системою.

Для користувачів із порушеннями опорно-рухового апарату характерні труднощі із використанням миші, трекпада чи сенсорного введення. Вони часто покладаються на клавіатурні комбінації, спеціальні перемикачі, голосове керування або інші альтернативні пристрої введення. У цьому випадку критично важливим стає мінімізація дрібних інтерфейсних компонентів та забезпечення оптимальної розкладки елементів. Погано структурована сторінка, дрібні кнопки, необхідність виконання точних рухів або складних жестів можуть суттєво знизити можливість ефективної навігації. Також важливо забезпечити достатні інтервали між інтерактивними елементами, можливість масштабування та спрощення інтерфейсу.

Когнітивні порушення формують окрему групу особливостей взаємодії. Користувачі з такими порушеннями можуть мати труднощі із сприйняттям складних текстів, багаторівневої навігації, перевантаженості інтерфейсу або швидко змінюваного контенту. Для них критичною є чіткість подання інформації, доступність мови, передбачуваність поведінки компонентів, а також відсутність відволікаючих анімацій та непотрібних візуальних ефектів. Когнітивна доступність передбачає використання зрозумілих структур, шаблонів, підказок, а також адаптивних механізмів, які дозволяють спростити інтерфейс або підлаштувати його під індивідуальний темп взаємодії.

Окремо слід відзначити вплив контексту використання, що включає тип пристрою, освітленість, рівень цифрової грамотності, фізичний стан користувача або допоміжні технології, якими він користується. Навіть користувач без діагностованих порушень може тимчасово опинитися в умовах,

які імітують обмеження доступності, наприклад, при яскравому сонячному світлі, роботі на маленькому екрані або за умов обмеженої моторики. Це підтверджує універсальність підходів до доступності та їх користь для всього спектра користувачів.

Особливості взаємодії користувачів з обмеженими можливостями визначають вимоги до вебзастосунків на рівні архітектури, логіки інтерфейсу та технологічної реалізації. Забезпечення структури документа, що відповідає семантичним стандартам, можливість керування без миші, адаптивна візуальна складова, альтернативні форми подання контенту та інтерфейсна передбачуваність – це необхідні умови повноцінної взаємодії таких користувачів із вебресурсами. Розуміння цих особливостей є основою подальшого аналізу бар'єрів доступності та формування методів адаптації вебзастосунків [2].

## 1.2 Види бар'єрів доступності та їх класифікація

Забезпечення доступності вебзастосунків передбачає виявлення та усунення бар'єрів, які перешкоджають або ускладнюють взаємодію користувачів з обмеженими можливостями з вебінтерфейсами. Бар'єри доступності проявляються на різних рівнях структури вебзастосунку – від архітектури до візуального подання контенту – і можуть мати як технічний, так і когнітивний характер. Їх своєчасна ідентифікація є необхідною умовою побудови інклюзивних цифрових сервісів, відповідних міжнародним стандартам та рекомендаціям [3].

Загалом бар'єри доступності поділяються на чотири основні групи: візуальні, слухові, моторні та когнітивні. Кожна з цих груп відображає специфіку взаємодії користувача з вебресурсом та викликає певні технічні вимоги до інтерфейсу.

До візуальних бар'єрів належать обмеження, що виникають у користувачів із частковою або повною втратою зору, порушеннями кольоросприйняття та контрастності. Найпоширенішими проблемами є використання недостатнього контрасту між текстом і фоном, наявність дрібних шрифтів, відсутність можливості масштабування, а також застосування кольору як єдиного індикатора стану елемента. Важливим аспектом є відсутність альтернативного тексту для зображень, іконок та графічних елементів, через що скрінрідери не можуть коректно інтерпретувати контент. До цієї групи також належить неправильне використання структури заголовків та відсутність семантичної розмітки, що ускладнює навігацію для користувачів, які покладаються на читачі екрану.

Слухові бар'єри стосуються користувачів із частковою або повною втратою слуху. У більшості випадків вони виникають при роботі з мультимедійними елементами, що містять аудіоінформацію. Відсутність субтитрів або текстового дублювання для відео, використання звукових сигналів без візуальних індикаторів, а також відсутність інструкцій для взаємодії з аудіоелементами призводять до втрати важливої інформації. Навіть у випадках, коли відеоматеріали мають субтитри, вони можуть бути неповними або не відповідати змісту, що також погіршує доступність. З огляду на це сучасні рекомендації передбачають не лише обов'язкове додавання субтитрів, але й можливість керування аудіоелементами через клавіатуру та наявність дублювальної текстової інформації.

Моторні бар'єри виникають у користувачів із порушеннями опорно-рухового апарату або обмеженими можливостями точного керування мишкою чи сенсорним екраном. До таких бар'єрів належать дрібні інтерактивні елементи, тісне розташування кнопок, складні жести, що потребують значної точності або багатокрокових дій, а також відсутність повної клавіатурної навігації. Нерідко розробники приділяють недостатню увагу логіці фокусування або можливості оперувати елементами інтерфейсу без використання мишки, що унеможлиблює взаємодію з вебзастосунком для

великої частини користувачів. Також проблемами є відсутність механізмів уникнення повторюваних завдань для користувачів, яким важко виконувати тривалі або точні послідовності дій.

Когнітивні бар'єри стосуються користувачів із порушеннями сприйняття, уваги, пам'яті, логічного мислення чи частковими неврологічними особливостями. Такі користувачі можуть мати труднощі з розумінням складних інструкцій, обробленням великої кількості інформації, взаємодією з перевантаженим інтерфейсом або інтерфейсами, що мають непередбачувану поведінку. Бар'єрами стають неоднозначні повідомлення, невідповідність між діями користувача та реакцією системи, надмірне використання анімацій або динамічних елементів, які відволікають увагу. Для подолання цих бар'єрів важливим є використання простої та зрозумілої мови, а також можливість адаптувати інтерфейс: спростити меню, увімкнути лаконічний режим чи надати покрокові інструкції.

Окрему групу становлять технічні бар'єри, які зумовлені неправильним використанням вебтехнологій. Вони включають некоректну HTML-структуру, відсутність ARIA-атрибутів або їх неправильне застосування, погано реалізовану адаптивність, несумісність з асистивними технологіями та порушення принципів кросбраузерності. Технічні бар'єри часто виникають через недостатнє знання розробниками стандартів доступності або нехтування ними під час створення інтерфейсу.

Також варто виокремити контекстуальні бар'єри, які не пов'язані безпосередньо з фізичними чи когнітивними обмеженнями, але виникають у реальних умовах використання вебзастосунку. Прикладами є погане освітлення, низька якість Інтернет-з'єднання, використання мобільного пристрою з маленьким екраном або тимчасові обмеження моторики. Такі бар'єри підтверджують важливість універсального дизайну, який дозволяє враховувати потреби максимально широкого кола користувачів.

Таким чином, класифікація бар'єрів доступності дозволяє систематизувати проблеми, з якими стикаються користувачі з обмеженими

можливостями, і сформувати вимоги до проєктування вебінтерфейсів. Розуміння природи цих бар'єрів є основою для розроблення ефективних методів адаптації вебзастосунків, що забезпечать інклюзивність та повноцінну взаємодію з цифровими системами. У наступному підрозділі розглядаються міжнародні стандарти доступності, які визначають принципи та критерії створення таких інтерфейсів [4].

### 1.3 Міжнародні стандарти доступності

Міжнародні стандарти доступності відіграють ключову роль у формуванні підходів до розроблення вебзастосунків, які є інклюзивними та придатними для використання широким колом користувачів, зокрема людьми з обмеженими можливостями. Ці стандарти визначають принципи, критерії, вимоги та методичні рекомендації, що регламентують створення доступного контенту, структури вебсторінок і логіки взаємодії. Їх дотримання забезпечує відповідність інтерфейсів сучасним нормам цифрової інклюзивності й сприяє підвищенню якості користувацького досвіду [5].

Найважливішим і найпоширенішим стандартом є Web Content Accessibility Guidelines (WCAG), розроблений Консорціумом Всесвітньої павутини (W3C). WCAG представлений у декількох редакціях – WCAG 2.0, WCAG 2.1 та WCAG 2.2 – й охоплює принципи доступності, які можна узагальнити у чотирьох ключових категоріях: сприйнятність, керованість, зрозумілість та надійність. Принцип сприйнятності вимагає, щоб інформація та елементи інтерфейсу були представлені у формах, доступних для різних сенсорних можливостей. Керованість стосується забезпечення доступу до всіх функцій вебзастосунку за допомогою різних способів керування, включно з клавіатурою та альтернативними пристроями. Зрозумілість передбачає подання контенту та функціоналу у передбачуваний, логічній та легкій для інтерпретації формі. Принцип надійності орієнтований на сумісність із сучасними та

майбутніми асистивними технологіями. У рамках WCAG визначено три рівні відповідності – А, АА та ААА, кожен з яких передбачає різний ступінь доступності.

Іще одним значущим нормативом є Section 508 – частина законодавства Сполучених Штатів Америки, яке регламентує вимоги доступності для інформаційно-комунікаційних технологій федеральних установ. Section 508 орієнтований на забезпечення рівного доступу до цифрових ресурсів для людей з інвалідністю та визначає обов'язкові вимоги до розроблення вебзастосунків, апаратних засобів і програмних систем, що використовуються в державному секторі. На відміну від WCAG, який є рекомендаційним стандартом, Section 508 має юридичну силу в рамках США та значно впливає на практику впровадження доступності у державних і корпоративних системах [6].

У країнах Європейського Союзу головним нормативним документом є EN 301 549, розроблений Європейським інститутом стандартизації телекомунікацій (ETSI). Цей стандарт регулює вимоги доступності для ІКТ-продукції та послуг у межах публічних закупівель, що стимулює впровадження доступності в державних сервісах та комерційних системах. EN 301 549 тісно пов'язаний зі стандартами WCAG, але також містить додаткові вимоги, які стосуються апаратних пристроїв, програмного забезпечення, мобільних додатків та мультимедійних систем. Виконання вимог цього стандарту є обов'язковим для організацій, які беруть участь у державних тендерах у країнах ЄС, що стимулює систематичне впровадження доступності.

Окреме місце займає стандартизація на рівні протоколів та інтерфейсів, зокрема специфікація WAI-ARIA (Accessible Rich Internet Applications), розроблена в межах W3C Web Accessibility Initiative. ARIA дозволяє надавати додаткові семантичні атрибути HTML-елементам, що забезпечує коректну інтерпретацію компонентів інтерфейсу скрінрідерами та іншими асистивними технологіями. Використання ARIA особливо важливе у складних динамічних вебзастосунках, де компоненти створюються програмно або не мають відповідних нативних HTML-ролей. Проте застосування ARIA вимагає

обережності, оскільки неправильна розмітка може призвести до погіршення доступності, а не її покращення [7].

У контексті мультимедійного контенту застосовується стандарт Timed Text Markup Language (TTML), що визначає структуру субтитрів і текстових альтернатив для відеоматеріалів. Завдяки таким стандартам забезпечується можливість інтеграції доступного медіаконтенту у відеоплатформи, освітні вебзастосунки та інформаційні ресурси.

Варто також відзначити міжнародну тенденцію до гармонізації стандартів. Багато державних органів та приватних компаній у світі приймають WCAG як основний норматив, а EN 301 549 та Section 508 частково або повністю базуються на ньому. Це дозволяє створювати уніфіковані підходи до оцінювання доступності та забезпечує сумісність між різними ринками й екосистемами цифрових продуктів.

Міжнародні стандарти доступності формують комплексний нормативний базис, який визначає вимоги до створення інклюзивних вебзастосунків. Їх дотримання не лише підвищує якість взаємодії користувачів з обмеженими можливостями, а й забезпечує відповідність продуктів законодавчим нормам, стандартам галузі та вимогам ринку. У наступному підрозділі розглянуто технології асистивного доступу, що безпосередньо забезпечують взаємодію користувача з вебінтерфейсами [8].

#### 1.4 Технології асистивного доступу

Технології асистивного доступу відіграють важливу роль у забезпеченні можливості взаємодії користувачів з обмеженими можливостями з вебзастосунками. Вони компенсують порушення сенсорних, моторних або когнітивних функцій та забезпечують альтернативні способи сприйняття інформації й управління інтерфейсом. Розуміння принципів роботи асистивних

технологій є необхідною умовою для створення вебінтерфейсів, сумісних із сучасними засобами доступності.

Одними з найпоширеніших інструментів є технології екранного читання (screen readers). Такі програми, як JAWS, NVDA, VoiceOver або TalkBack, аналізують структуру вебсторінки та озвучують її користувачу або перетворюють інформацію у шрифт Брайля за допомогою брайлівських дисплеїв. Ефективність роботи скрінрідера залежить від коректності семантичної розмітки HTML, наявності ARIA-атрибутів, логічної структури заголовків та правильного використання елементів форм. Порушення цих принципів може призвести до того, що користувач отримає хаотичну, непослідовну або частково недоступну інформацію. Тому розробники мають приділяти особливу увагу правильній структуризації контенту та передбачуваності поведінки інтерфейсу [9].

Для користувачів зі зниженим зором важливими є технології збільшення зображення та екранні лупи, такі як ZoomText або вбудовані засоби масштабування в операційних системах. Вони забезпечують збільшення окремих частин екрану, підвищення контрастності або зміну кольорової гами. Для коректної роботи масштабування вебінтерфейси повинні мати адаптивну верстку, відносні одиниці вимірювання та механізми, що не обмежують масштаб контенту. Використання фіксованих значень або блокування зуму призводить до серйозних проблем доступності для значної частини користувачів.

У випадку порушень слуху найважливішими є системи візуалізації звукової інформації, включаючи субтитри, текстові альтернативи та візуальні індикатори. Наприклад, вебплеєри мають забезпечувати можливість виведення субтитрів, їх налаштування та відображення не тільки для відео, а й для аудіоконтенту у форматі транскрипту. У багатьох типових інтерфейсах звукові сигнали використовуються для повідомлення про помилки або системні події, тому такі сигнали повинні дублюватися візуальними повідомленнями. Асистивні рішення для нечуючих користувачів також можуть включати

апаратні пристрої, що синхронізуються з контентом, наприклад вібраційні індикатори або світлові сигналізатори.

Користувачі з порушеннями моторики часто використовують альтернативні пристрої введення, серед яких виділяють перемикачі, спеціальні клавіатури, джойстики, системи відстеження рухів голови або очей. Поширеним є також голосове керування, представлене у вигляді таких систем, як Dragon NaturallySpeaking, Voice Control на macOS або голосові команди в мобільних ОС. Для коректної взаємодії з такими пристроями вебінтерфейси повинні мати чітку логіку переходу між елементами, відповідні ролі ARIA та можливість повноцінного керування через клавіатуру. Складні жести, дрібні кнопки або анімації, що вимагають точних рухів, стають непереборними бар'єрами для таких користувачів.

Для користувачів із когнітивними порушеннями важливими є технології спрощеного подання інформації, системи структурованих підказок, автоматичне читання тексту та інструменти підтримки уваги. До них також належать інтерфейси, що дозволяють увімкнути режим з мінімальною кількістю елементів, зменшити кількість візуальних ефектів або використовувати зрозумілі ілюстрації. Асистивні програми можуть пропонувати альтернативні формати подання інформації, спрощені інтерфейси або інструменти виділення ключових елементів сторінки.

Важливу роль у забезпеченні доступності відіграють браузерні та операційні засоби, такі як режим високої контрастності в Windows, доступність на рівні macOS, функції Android Accessibility Suite або універсальні налаштування iOS [10]. Ці засоби надають механізми масштабування, озвучування, керування жестами, інверсії кольорів, спрощення анімації тощо. Сучасні вебзастосунки повинні бути розроблені таким чином, щоб не конфліктувати з цими механізмами та коректно адаптуватися до змінених системних налаштувань.

Технології асистивного доступу формують комплексний інструментарій, що дозволяє користувачам з обмеженими можливостями повноцінно

взаємодіяти з вебзастосунками. Для забезпечення їх ефективної роботи необхідно враховувати принципи доступного дизайну, дотримуватися стандартів та забезпечувати сумісність інтерфейсів із сучасними асистивними системами.

### 1.5 Методи забезпечення доступності в сучасних вебзастосунках

Методи забезпечення доступності у сучасних вебзастосунках охоплюють широкий комплекс технічних, архітектурних та дизайнерських рішень, що спрямовані на усунення бар'єрів для користувачів з обмеженими можливостями. Ці методи базуються на міжнародних стандартах доступності, принципах універсального дизайну, особливостях взаємодії з асистивними технологіями та сучасних практиках розробки інтерактивних інтерфейсів.

Одним із ключових підходів є забезпечення семантичної коректності структури документа [11]. Правильне використання HTML-елементів, побудова ієрархії заголовків, групування форм, позначення ролей та станів інтерфейсних компонентів дозволяють асистивним технологіям коректно інтерпретувати зміст сторінки. Семантика забезпечує доступність незалежно від візуального дизайну та є критичним фактором для користувачів, які покладаються на скрінрідери. У складних інтерфейсах застосовуються ARIA-атрибути, що уточнюють призначення елементів, але вони повинні використовуватися лише як доповнення, а не заміна стандартних HTML-компонентів [12].

Важливим методом є забезпечення доступності клавіатурної навігації, що дозволяє користувачам взаємодіяти з вебзастосунком без використання миші. Це включає реалізацію логічного порядку переходу між елементами (tab order), підсвічування фокуса, можливість керування всіма інтерактивними компонентами за допомогою клавіатури, а також підтримку стандартних комбінацій клавіш. Важливо, щоб фокус не блокувався на окремих елементах, а модальні вікна мали механізми коректної обробки фокусування. Клавіатурна

доступність є критичною не лише для користувачів із моторними порушеннями, а й для тих, хто використовує альтернативні пристрої введення.

Значну роль відіграють методи, пов'язані з адаптивністю та масштабованістю інтерфейсу. Використання відносних одиниць вимірювання, адаптивних сіток і флексбоксів дозволяє забезпечити коректне відображення елементів при збільшенні масштабу. Здатність інтерфейсу зберігати функціональність при масштабуванні на 200–400 % є однією з вимог WCAG. Адаптивний дизайн також передбачає можливість коректної взаємодії на різних типах пристроїв, що є важливим не лише для користувачів із вадами зору, але й для всіх, хто працює з обмеженим простором екрану [13].

Одним із важливих методів є забезпечення достатнього контрасту та можливості візуальної адаптації. Правильний підбір кольорової гами, контрастність між текстом і фоном, відображення активних станів елементів, можливість перемикання режимів (наприклад, «темна тема», «високий контраст») істотно впливають на сприйняття інформації користувачами зі зниженим зором. Важливо уникати використання кольору як єдиного носія інформації — усі критичні зміни стану, помилки або попередження повинні дублюватися текстовими або графічними маркерами.

Окреме місце займають методи забезпечення доступності мультимедійного контенту. До них належать додавання субтитрів до відео, транскриптів до аудіозаписів, можливість керування мультимедіа з клавіатури, а також уникнення автоматичного відтворення аудіо. WCAG рекомендує забезпечувати спрощені механізми зупинки, паузи та регулювання гучності, оскільки неконтрольований мультимедійний контент утворює додаткові бар'єри для людей із порушеннями слуху та когнітивними труднощами [14].

Важливими є також методи, пов'язані з логічною структурою та зрозумілістю інтерфейсу. Просте подання інформації, передбачувана навігація, зрозумілі підказки та інструкції, однозначні повідомлення про помилки та стан системи — необхідні для користувачів із когнітивними порушеннями. У цьому контексті важливими є принципи «plain language», простих шаблонів, а також

можливість створення адаптованого режиму інтерфейсу, що знижує кількість елементів і відволікаючих ефектів.

Додатково застосовуються методи перевірки доступності, які включають ручне тестування, автоматизовані інструменти (axe, Lighthouse, WAVE), а також тестування із залученням осіб з обмеженими можливостями. Автоматизовані засоби дозволяють виявити найбільш поширені помилки, такі як відсутність альтернативного тексту або некоректну структуру заголовків, але не здатні оцінити повноцінність користувацького досвіду. Тому найефективнішим є комбінований підхід, який включає кілька методів оцінювання [15].

Методи забезпечення доступності формують багатовимірний комплекс рішень, що охоплює семантику, візуальне подання, навігацію, мультимедійний контент, адаптивність та логічну структуру інтерфейсу. Їх застосування є основою для створення інклюзивних вебзастосунків, здатних задовольнити потреби широкого кола користувачів.

## 1.6 Аналіз існуючих підходів, досліджень і програмних рішень

Розвиток вебдоступності сформував широкий спектр підходів та технологічних рішень, які спрямовані на забезпечення інклюзивності цифрових платформ. Дослідження в цій галузі охоплюють як теоретичні моделі доступності, так і практичні інструменти адаптації вебінтерфейсів, а також засоби автоматизованого перевіряння доступності. Аналіз сучасних робіт дозволяє визначити ключові напрями розвитку та встановити їхні переваги й недоліки з погляду ефективності, універсальності та сумісності з асистивними технологіями [16].

Одним із найбільш усталених підходів є модельна доступність, що зорієнтована на відповідність вебзастосунку міжнародним стандартам WCAG. У цьому випадку основна увага приділяється формальній відповідності

критеріям рівня А або АА, що передбачає дотримання вимог до контрастності, семантики, клавіатурної навігації та мультимедійних альтернатив. Значна кількість робіт у цій сфері фокусується на формалізації критеріїв доступності, методиках оцінки відповідності стандартам та розробленні інструментів для виявлення порушень. Хоча такий підхід є фундаментальним, він часто не враховує індивідуальних потреб користувачів і не забезпечує гнучкої адаптації інтерфейсу під конкретний профіль взаємодії.

Іншим поширеним напрямом є інтерфейсна адаптація, що передбачає зміну структури та поведінки інтерфейсу відповідно до потреб користувача. У наукових роботах цей підхід реалізується за допомогою механізмів адаптивного дизайну, налаштовуваних користувацьких профілів, зміни масштабів, кольорових схем або спрощення структури. Дослідження у цій сфері показують ефективність адаптивних компонентів, здатних змінюватися в реальному часі залежно від типу порушення або контексту використання. Проте значна частина таких рішень має обмежений набір налаштувань або не забезпечує автоматичної адаптації без участі користувача [17].

Важливим напрямом є розвиток інтелектуальних систем виявлення проблем доступності, які базуються на алгоритмах машинного навчання. Такі системи здатні аналізувати вебінтерфейси, виявляти потенційні проблеми доступності та пропонувати шляхи їх усунення. Дослідження вказують, що машинне навчання може застосовуватися для семантичного аналізу структури сторінки, прогнозування поведінки користувачів та ранжування елементів інтерфейсу за важливістю. Проте професійне застосування таких систем обмежене складністю формування навчальних вибірок і необхідністю взаємодії з різними типами вебінтерфейсів.

У практичній площині набули поширення численні інструменти автоматизованої перевірки доступності, серед яких ахе DevTools, WAVE Evaluation Tool, Lighthouse, Accessibility Insights тощо. Ці інструменти дозволяють виявляти найбільш поширені помилки, такі як відсутність альтернативного тексту, неправильна структура заголовків, проблеми з

контрастністю або клавіатурною навігацією [18]. Проте автоматизовані засоби не можуть повністю замінити ручне тестування або оцінку досвіду користувачів, оскільки вони не здатні визначити контекст, доступність складної динамічної поведінки елементів або оцінити інтерфейс з позиції когнітивного навантаження.

Серед практичних рішень окрему групу становлять готові модулі доступності, які інтегруються у вебзастосунки як сторонні плагіни. Це, наприклад, системи доступності у вигляді «допоміжних панелей», які дозволяють користувачу вмикати високий контраст, збільшувати шрифти, змінювати інтерліньяж, вимикати анімацію або активувати режим читання. Попри простоту інтеграції й швидкий ефект, ці рішення часто мають низку недоліків: не завжди враховують індивідуальний контекст користувача, можуть бути несумісними з асистивними технологіями, а інколи навіть погіршують доступність через дублювання функціональності або конфлікти з системними налаштуваннями.

Сучасні дослідження також фокусуються на побудові персоналізованих моделей доступності, що передбачають створення окремих профілів користувачів з конкретними порушеннями. Такі профілі можуть містити інформацію про потрібний рівень масштабування, допустиму контрастність, темп взаємодії, необхідність дублювання елементів або спрощення структури. Персоналізація дозволяє створити більш гнучке середовище, яке адаптується до реальних потреб людини, а не лише до формальних вимог стандартів. Проте подібні системи потребують механізмів збору й аналізу користувацьких даних, що створює додаткові вимоги до безпеки та конфіденційності [19].

Аналіз існуючих досліджень свідчить, що попри значний прогрес, залишається низка невирішених проблем. Більшість підходів фокусуються або на автоматизації перевірки доступності, або на формальній відповідності стандартам, але не забезпечують комплексної адаптації інтерфейсу під конкретні потреби користувачів. Інструменти інтелектуального аналізу поки що не мають достатньої точності в широкому спектрі інтерфейсів, а інтегровані

панелі доступності часто працюють ізольовано від загальної логіки вебзастосунку.

Існуючі підходи забезпечують базовий рівень доступності, але не формують повноцінної інструментарію для динамічної адаптації вебінтерфейсів. Це відкриває можливість для розроблення нових методів, орієнтованих на персоналізацію, гнучкість і глибоку інтеграцію в архітектуру вебзастосунків. У наступному підрозділі буде сформульовано постановку задачі дослідження, що окреслює напрями роботи в контексті зазначених недоліків.

### 1.7 Постановка задачі дослідження

Таким чином, адаптація вебзастосунків для користувачів з обмеженими можливостями є актуальним завданням. Прийнято рішення щодо розроблення програмного застосунку із адаптацією для користувачів з обмеженими можливостями.

Об'єктом дослідження є процес адаптації вебзастосунків до потреб користувачів з обмеженими можливостями.

Метою дослідження є розроблення та обґрунтування методу адаптації вебзастосунку для користувачів з обмеженими можливостями, який забезпечить підвищення доступності та якості взаємодії за рахунок поєднання семантичних, інтерфейсних і персоналізованих механізмів.

Для досягнення мети необхідно вирішити такі завдання:

- провести формалізацію функціональних та нефункціональних вимог до адаптивного вебінтерфейсу з урахуванням різних типів порушень;
- дослідити методи модифікації інтерфейсних компонентів, включаючи зміни візуальних параметрів, структури навігації та представлення контенту;
- розробити модель персоналізації інтерфейсу, що дозволяє створювати профілі користувачів з індивідуальними параметрами доступності;

- запропонувати архітектурне рішення та програмну реалізацію методу адаптації;
- здійснити експериментальну оцінку ефективності запропонованого методу та порівняти його результати з існуючими рішеннями.

## 2 МЕТОДИ, МОДЕЛІ ТА ТЕХНОЛОГІЧНІ ПІДХОДИ ДО АДАПТАЦІЇ ВЕБЗАСТОСУНКІВ

### 2.1 Формалізація вимог до адаптивності і доступних вебінтерфейсів

Формалізація вимог до адаптивності та доступності вебінтерфейсів є ключовим етапом у створенні вебзастосунків, орієнтованих на потреби користувачів з обмеженими можливостями. На цьому етапі визначаються критерії, правила та обмеження, які мають бути враховані під час проектування інтерфейсу, структури контенту та механізмів взаємодії. Формалізація вимог забезпечує системний підхід до створення доступного цифрового продукту, узгоджує очікування користувачів із можливостями технологій та дає змогу оцінити якість реалізації адаптивних рішень.

Першою групою вимог є функціональні вимоги, що стосуються безпосередньої взаємодії користувача з вебінтерфейсом. До них належать можливість повноцінного керування інтерфейсом за допомогою різних способів введення (клавіатури, голосових команд, перемикачів), доступ до всіх функцій незалежно від сенсорних або моторних обмежень, а також забезпечення коректної поведінки інтерфейсних компонентів у динамічних сценаріях. Функціональні вимоги також охоплюють реалізацію адаптивних механізмів зміни розміру тексту, масштабування інтерфейсу, налаштування кольорової схеми та можливість увімкнення спрощеного режиму інтерфейсу. До цієї ж групи належить підтримка альтернативних шляхів доступу до контенту – наприклад, дублювання мультимедіа текстом або забезпечення семантичної структури, що дозволяє скрінрідеру коректно інтерпретувати логіку сторінки [20].

Другою групою є нефункціональні вимоги, які визначають загальну якість вебінтерфейсу. Вони включають вимоги доступності (відповідність WCAG 2.x на рівні А чи AA), вимоги до продуктивності, надійності, сумісності та безпеки.

Важливими є вимоги, пов'язані зі стійкістю інтерфейсу при зміні масштабу, роботі в умовах низької швидкості Інтернету, сумісності з асистивними технологіями і різними браузерами. Нефункціональні вимоги також охоплюють передбачуваність поведінки інтерфейсних компонентів, логічність навігації, узгодженість візуального оформлення та мінімізацію когнітивного навантаження.

Особливе місце у формалізації вимог займають семантичні вимоги, що регламентують правильну структуру HTML-документа. Ці вимоги охоплюють ієрархію заголовків, правильне застосування елементів `<nav>`, `<main>`, `<section>`, `<article>`, семантичне розміщення кнопок і посилань, коректне визначення станів і ролей за допомогою ARIA-атрибутів. Семантичні вимоги є критичними для сумісності з екранними читачами та іншими асистивними технологіями, оскільки саме семантичне розташування елементів визначає можливість озвучування, навігації та інтерпретації структури сторінки [21].

До важливих вимог належать також візуальні вимоги, що забезпечують можливість комфортного сприйняття контенту. Вони передбачають дотримання мінімальних значень контрастності між текстом і фоном, оптимальний розмір шрифту, достатні інтервали між елементами, можливість обмеження анімацій і миготіння, а також послідовне та зрозуміле використання кольорів. Візуальні вимоги включають здатність інтерфейсу адаптуватися до системних налаштувань, таких як режим високої контрастності або інверсія кольорів.

Не менш важливою групою є вимоги до навігації та управління, які визначають логіку переходів між елементами, структуру меню, передбачуваність інтерфейсних дій та доступність усіх компонентів за допомогою клавіатури. Сюди належать вимоги до фокусування, уникнення пасток фокуса, забезпечення інформативних підказок і чітких описів полів форми.

У межах адаптивності формулюються вимоги до персоналізації, тобто можливості створення індивідуального профілю користувача, який визначає необхідні параметри інтерфейсу [22]. Персоналізація передбачає гнучкість

системи, що дозволяє змінювати масштаб шрифтів, перемикає теми, активувати режим спрощеної структури або налаштувати швидкість анімацій. Важливим аспектом є можливість збереження цих налаштувань та їх автоматичного застосування під час повторної взаємодії користувача з вебзастосунком.

Методи забезпечення доступності формують багатовимірний комплекс рішень, що охоплює семантику, візуальне подання, навігацію, мультимедійний контент, адаптивність та логічну структуру інтерфейсу. Їх застосування є основою для створення інклюзивних вебзастосунків, здатних задовольнити потреби широкого кола користувачів.

## 2.2 Методи модифікації фронтенду під потреби користувачів з обмеженими можливостями

Модифікація фронтенду під потреби користувачів з обмеженими можливостями з ключовим напрямом адаптації вебзастосунків, що забезпечує інклюзивний доступ до цифрового контенту. Оскільки вебінтерфейс виступає основним середовищем взаємодії користувача з системою, саме на рівні фронтенду реалізуються механізми візуального представлення інформації, навігації, доступу до функцій та динамічної інтерактивності. Методи модифікації фронтенду спрямовані на зміну параметрів візуального оформлення, структури навігації та інтерактивних елементів, що дозволяє компенсувати сенсорні, моторні або когнітивні обмеження користувачів.

### 2.2.1 Адаптація візуальної складової

Адаптація візуальної складової є однією з найпоширеніших практик забезпечення доступності вебінтерфейсів. Користувачі зі зниженим зором часто

мають потребу у збільшенні тексту, підвищенні контрастності, зміні кольорових схем, спрощенні графічних елементів та обмеженні динамічних ефектів. Тому система адаптації фронтенду повинна забезпечувати можливість налаштування цих параметрів інтерфейсу.

Одним із базових методів є зміна масштабування тексту та елементів інтерфейсу. Використання відносних одиниць (em, rem, vh, vw) дає змогу збільшувати або зменшувати розмір шрифтів і блоків без порушення структури сторінки. Важливо, щоб масштабування до 200–400 %, рекомендоване WCAG, не призводило до перекривання елементів або втрати інформації [23].

Іншим важливим методом є адаптація контрасту. Для користувачів з порушеннями кольоросприйняття критично важливо дотримуватися мінімальних значень контрастності: 4.5:1 для звичайного тексту та 3:1 для великого. Адаптивні модулі можуть дозволяти перемикати інтерфейс у режим високої контрастності або темну тему, що знижує навантаження на зір та покращує читабельність.

Також застосовуються методи зміни кольорової палітри для користувачів із дальтонізмом. Використання функцій CSS-фільтрів, спеціальних палітр або попередньо підготовлених тем дозволяє уникнути кольорової інформації, яку користувач може не сприймати коректно.

Важливими є й методи контролю динамічних елементів. Анімації, паралакс-ефекти та автоматичні переходи можуть негативно впливати на користувачів з вестибулярними або когнітивними порушеннями. Модифікація фронтенду повинна забезпечувати можливість вимикання анімації (prefers-reduced-motion) або уповільнення динамічних переходів [24].

Адаптація візуальної складової створює умови для комфортного сприйняття інформації та зменшує навантаження на зорову систему користувача.

### 2.2.2 Адаптація навігації та структури інтерфейсу

Адаптація навігації охоплює методи, що забезпечують предбачуваність, логічність і структурованість взаємодії з вебзастосунком. Для користувачів із моторними або когнітивними порушеннями складна навігація, багаторівневі меню або невпорядкована структура можуть стати суттєвим бар'єром.

Одним із ключових методів є реалізація повноцінної клавіатурної навігації. Усі елементи інтерфейсу повинні бути досяжними через клавішу Tab, а порядок фокусування має відповідати логічній структурі сторінки. Забезпечення видимого фокуса є критично важливим, оскільки дозволяє користувачеві орієнтуватися у поточному контексті [25].

Важливим аспектом адаптації навігації є структура меню. Спрощені або «плоскі» меню, розташовані у передбачуваних зонах інтерфейсу, значно полегшують взаємодію. У багатьох випадках корисним є впровадження «skip to content» посилань, що дозволяють пропустити повторювані навігаційні блоки.

Методи адаптації структури інтерфейсу також включають можливість перемикання інтерфейсу в режим спрощеної структури – тобто видалення другорядних або графічно складних елементів, зменшення кількості рівнів вкладеності та уникнення прихованих меню.

Для користувачів із моторними порушеннями значну роль відіграє оптимізація розмірів інтерактивних елементів. Кнопки, чекбокси, слайдери та інші елементи повинні мати мінімальний рекомендований розмір (44×44 px за WCAG), а також достатній простір між ними.

Не менш важливим методом є забезпечення модульності та предбачуваності інтерфейсу. Користувачі повинні отримувати однакову структуру та логіку керування на кожній сторінці або в кожній частині застосунку. Це дозволяє уникнути плутанини і знижує когнітивне навантаження.

### 2.2.3 Адаптація контенту та інтерактивних елементів

Контент і інтерактивні елементи є основними носіями інформації та функціональності вебзастосунку, тому їх адаптація є необхідною умовою доступності [26]. Методи адаптації контенту спрямовані на забезпечення зрозумілості, передбачуваності та структурованості подання інформації.

Одним із важливих підходів є використання простих мовних конструкцій і уникнення складних граматичних зворотів. Це особливо важливо для користувачів із когнітивними порушеннями. У деяких випадках може застосовуватися адаптивний «режим простого тексту», який видаляє складні елементи, скорочує текст або подає ключову інформацію у вигляді списків.

Для користувачів з порушеннями зору ключовим є забезпечення текстових альтернатив для зображень, іконок, графічних кнопок, діаграм та відео. Використання атрибутів `alt`, `aria-label`, `aria-describedby` та коректного підписування медіа забезпечує сумісність зі скрінрідерами.

Інтерактивні елементи повинні мати чіткі та однозначні стани. Активний, заблокований, вибраний або фокусований стан має бути візуально і семантично позначений. Це дозволяє користувачам однозначно інтерпретувати реакцію системи на їхні дії [27].

Складні компоненти, такі як модальні вікна, випадаючі списки, слайдери, каруселі та інтерактивні графіки, повинні мати підтримку клавіатури та коректну ARIA-розмітку. Наприклад, модальні вікна повинні блокувати фокус поза межами модалі та дозволяти закриття за допомогою клавіш.

Для мультимедійного контенту застосовуються методи дублювання інформації: субтитри для відео, транскрипти для аудіо, описові титри для складних графіків та діаграм [28].

Адаптація контенту та інтерактивних компонентів забезпечує доступність як для користувачів із сенсорними порушеннями, так і для тих, хто має труднощі у сприйнятті або виконанні складних дій.

### 2.3 Підходи до автоматичного виявлення проблем доступності

Автоматичне виявлення проблем доступності є важливою складовою забезпечення якісного та інклюзивного вебінтерфейсу. Сучасні вебзастосунки характеризуються складною структурою, динамічними компонентами та значною кількістю інтерактивних елементів, що робить ручне тестування доступності трудомістким і потенційно неповним. Тому у практиці розроблення широко застосовуються автоматизовані та напівавтоматизовані інструменти аналізу, які дозволяють прискорити процес перевірки, стандартизувати виявлення найпоширеніших помилок і забезпечити базову оцінку відповідності вимогам WCAG.

Одним із найбільш розповсюджених підходів є правилорегламентоване сканування інтерфейсу, коли інструмент перевіряє вебсторінку на наявність порушень, порівнюючи її структуру та стиль із набором визначених правил. Такі правила зазвичай включають вимоги до наявності атрибутів alt для зображень, достатньої контрастності між текстом і фоном, правильного порядку заголовків, доступності елементів форм, коректного фокусування та відсутності перешкод для клавіатурної навігації. Цей підхід реалізовано у таких інструментах, як ахе DevTools, Lighthouse, WAVE, Accessibility Insights. Його основною перевагою є висока швидкість та можливість автоматизації у процесі CI/CD, що дозволяє контролювати доступність на кожному етапі розробки [29].

Однак правилорегламентоване тестування має суттєві обмеження. Воно виявляє лише ті помилки, які можуть бути чітко формалізовані у вигляді правил. Інструменти не оцінюють контекст використання, не перевіряють логіку взаємодії та не здатні визначити, чи є інтерфейс інтуїтивно зрозумілим або когнітивно доступним. Наприклад, коректне наявне значення alt може бути змістовно некоректним, складні меню можуть бути формально доступними, але фактично незручними, а текст може бути граматично правильним, але складним для сприйняття. Таким чином, автоматизоване тестування є

ефективним лише на початковому етапі і потребує доповнення ручним аналізом.

Другим напрямом є аналіз поведінки інтерфейсу у динамічних сценаріях, що включає тестування компонентів, які створюються або змінюються за допомогою JavaScript. У сучасних односторінкових застосунках (SPA) значна частина інтерфейсу формується динамічно, а традиційні інструменти можуть не виявити помилки, які з'являються під час взаємодії користувача з елементами. Для цього використовуються інструменти, здатні виконувати симуляцію користувацьких дій, наприклад, Puppeteer, Playwright, Cypress у поєднанні з модулями перевірки доступності (axe-core) [30]. Такі підходи дозволяють автоматично перевіряти стан інтерфейсу після відкриття модальних вікон, перемикання вкладок, розгортання списків або зміни зазначених атрибутів.

Третім напрямом є застосування методів машинного навчання для оцінки доступності. У наукових роботах описано підходи, спрямовані на семантичний аналіз структури сторінки, виявлення невідповідностей між ролями елементів і їхньою поведінкою, визначення проблем читабельності або передбачуваності інтерфейсу. Такі системи можуть аналізувати макет сторінки як зображення або як дерево DOM і виявляти потенційні проблеми, які важко формалізувати правилами. Перевагою ML-підходів є здатність до узагальнення, а недоліком — потреба у великих навчальних вибірках та відсутність сучасних промислових рішень, придатних для широкого використання у розробці.

Четвертим поширеним підходом є напівавтоматичне тестування, коли інструмент пропонує розробнику або тестувальнику рекомендації щодо перевірки окремих елементів вручну. Наприклад, інструмент може повідомити, що елемент потенційно недоступний, але потребує ручної перевірки, або що контент може бути складним для сприйняття, але це не може бути визначено автоматично. Таким чином, напівавтоматичні засоби забезпечують баланс між формальним аналізом і людською оцінкою [31].

Окрім інструментів, що працюють локально або у браузері, існують хмарні сервіси перевірки доступності, які здійснюють автоматичну оцінку

сайтів без необхідності їхнього розгортання локально. Це можуть бути інтегровані інструменти для великих вебплатформ або корпоративні рішення для оцінки доступності у великих системах.

Автоматичне виявлення проблем доступності є важливою складовою забезпечення якісного користувацького досвіду, однак жоден із підходів не може повністю замінити ручне тестування. Автоматизація дозволяє виявити значну частину технічних порушень, проте оцінка когнітивної доступності, інтерпретації тексту, логічної структури та поведінки інтерфейсу все ще потребує участі спеціаліста. Саме тому сучасні системи адаптації вебзастосунків повинні поєднувати автоматизовані методи перевірки з гнучкими підходами до персоналізації та адаптивності, що є предметом подальших підрозділів.

#### 2.4 Моделі персоналізації вебінтерфейсів на основі профілю користувача

Персоналізація вебінтерфейсів є ключовим механізмом забезпечення адаптації вебзастосунків до індивідуальних потреб користувачів з обмеженими можливостями. На відміну від загальних підходів до доступності, які передбачають універсальні рекомендації для широкої аудиторії, персоналізація дозволяє створити інтерфейс, що динамічно змінюється відповідно до специфічних характеристик окремого користувача. Моделі персоналізації базуються на формуванні профілю користувача, який містить інформацію про його особливості, обмеження, переваги та типові сценарії взаємодії.

Першим типом моделей є декларативні моделі персоналізації. Вони ґрунтуються на явному зазначенні користувачем своїх потреб [32]. У таких системах користувач самостійно обирає необхідні параметри доступності: розмір шрифту, кольорову схему, швидкість анімацій, спосіб подання контенту, спрощений режим інтерфейсу тощо. Перевагою декларативного підходу є простота реалізації та висока відповідність реальним потребам користувача.

Водночас цей підхід вимагає від користувача певних технічних знань або здатності самотійно визначити свої потреби, що може бути складним для деяких категорій користувачів.

Другим типом є поведінкові моделі персоналізації, які ґрунтуються на автоматичному аналізі взаємодії користувача з інтерфейсом. У межах цього підходу система відстежує типові дії користувача, швидкість виконання операцій, складнощі у навігації, частоту помилок та інші метрики. На основі цих даних формується профіль, що описує індивідуальні труднощі або переваги користувача. Наприклад, якщо система визначає, що користувач повільно пересувається між елементами або пропускає дрібні компоненти, інтерфейс може автоматично збільшити інтерактивні елементи або активувати режим високої контрастності [33]. Перевагою такого підходу є відсутність необхідності у прямому втручанні користувача, однак він потребує складних механізмів збору та обробки даних, а також вирішення питань конфіденційності.

Третім типом є контекстні моделі персоналізації, які враховують зовнішні умови використання вебзастосунку. До контекстних параметрів належать налаштування операційної системи, характеристика пристрою, час доби, рівень освітлення, тип підключення, присутність адаптивних технологій. Наприклад, якщо система виявляє, що користувач активував режим високої контрастності у своїй операційній системі, вебзастосунок може автоматично застосувати відповідну кольорову схему. Контекстний підхід дозволяє забезпечити плавний і непомітний для користувача перехід до комфортного режиму взаємодії.

Окрему категорію становлять гібридні моделі персоналізації, які поєднують декларативні, поведінкові та контекстні підходи. У таких моделях система формує комплексний профіль користувача, який оновлюється динамічно під час взаємодії. Гібридні моделі є найбільш ефективними, оскільки дозволяють компенсувати недоліки окремих підходів та забезпечити високу точність адаптації. Наприклад, користувач може вручну задати початкові

параметри інтерфейсу, а система надалі коригуватиме їх, враховуючи поведінкові особливості та контекст.

Важливим елементом підходів до персоналізації є структура профілю користувача. У загальному випадку профіль складається з атрибутів, що описують стан зору, слуху, моторики, когнітивних функцій, а також індивідуальні параметри комфорту, такі як бажаний шрифт, контраст, спосіб навігації чи доступ до альтернативного контенту. Профіль може зберігатися локально у браузері або на сервері, що дозволяє забезпечити сталість налаштувань на різних пристроях [34].

Моделі персоналізації також можуть включати механізми пріоритезації правил, коли окремі налаштування мають вищий рівень важливості. Наприклад, для користувача з важкими порушеннями зору збільшення шрифту та високий контраст мають вищий пріоритет, ніж декоративні параметри. Це дозволяє системі уникнути конфліктів між правилами та забезпечити максимальний комфорт [35].

Моделі персоналізації вебінтерфейсів на основі профілю користувача надають можливість створення інтерфейсу, що враховує індивідуальні особливості користувача, забезпечує гнучкість та адаптивність і значно підвищує доступність.

## 2.5 Використання машинного навчання для покращення адаптації інтерфейсів

Застосування методів машинного навчання у сфері адаптації вебінтерфейсів відкриває нові можливості для створення інтелектуальних систем, здатних автоматично підлаштовуватися під потреби користувачів з обмеженими можливостями. На відміну від традиційних підходів, де адаптація зводиться до статичних правил або ручних налаштувань, машинне навчання

дозволяє аналізувати поведінку користувача, передбачати його потреби та пропонувати персоналізовані рішення у динамічному режимі [36].

Одним із ключових напрямів застосування машинного навчання є аналіз поведінкових даних, що дозволяє визначати характерні труднощі користувача. Сучасні вебзастосунки здатні фіксувати час виконання окремих дій, частоту помилок, використання клавіатурних скорочень, швидкість навігації, пропущені або важкодоступні елементи. На основі цих даних модель може класифікувати тип можливих обмежень, наприклад, уповільнене виконання рухових операцій, складнощі зі сприйняттям дрібного тексту або труднощі у навігації складними меню. Такий аналіз дозволяє системі автоматично рекомендувати оптимальні параметри інтерфейсу або навіть застосовувати їх без участі користувача [37].

Машинне навчання може застосовуватись для аналізу структури сторінки з метою виявлення потенційних проблем доступності. Використання алгоритмів комп'ютерного зору дає змогу аналізувати візуальні компоненти інтерфейсу та визначати, наскільки вони відповідають принципам зручності сприйняття. Наприклад, модель може автоматично оцінити контрастність тексту, читабельність елементів або структуру розмітки. Алгоритми класифікації та сегментації можуть визначити, чи є групування елементів логічним або чи може користувач зі зниженим зором ідентифікувати важливі елементи.

Окремим напрямом є використання методів рекомендаційних систем. У таких системах профілі користувачів формуються на основі спільних ознак або схожих сценаріїв взаємодії. Якщо система виявляє, що певна група користувачів із подібними обмеженнями обирає схожі налаштування інтерфейсу, вона може рекомендувати ці параметри новим користувачам з аналогічними характеристиками. Це дозволяє створити більш точну й обґрунтовану модель адаптації, яка враховує колективний досвід та реальну ефективність різних варіантів налаштувань.

Методи глибинного навчання також можуть застосовуватися для прогнозування комфортних параметрів користувача. Наприклад, рекурентні нейронні мережі або трансформери можуть аналізувати послідовність дій користувача у вебінтерфейсі та робити висновки щодо того, які елементи викликають труднощі, яка швидкість навігації є типовою та які параметри інтерфейсу можуть спростити взаємодію [38]. Такі моделі здатні працювати у режимі онлайн-адаптації, що означає динамічне оновлення рекомендацій залежно від поточної поведінки користувача.

Важливим аспектом використання машинного навчання є етичні й технічні обмеження. Для ефективного навчання моделей необхідні великі масиви даних, у тому числі дані про користувачів з різними порушеннями. Це створює потребу у захисті персональних даних, забезпеченні анонімності та дотриманні регуляторних вимог. Крім того, моделі повинні уникати упередженості й помилкової класифікації, яка може призвести до некоректної адаптації інтерфейсу.

Попри складність, застосування машинного навчання має значний потенціал. Воно дозволяє відійти від універсальних рішень і перейти до персоналізованих підходів, що враховують динамічний характер взаємодії користувача з вебінтерфейсом. У результаті користувач отримує більш точні, доречні та ефективні налаштування, а система стає інклюзивнішою та зручнішою.

Машинне навчання розширює можливості адаптації вебінтерфейсів, забезпечуючи інтелектуальну підтримку користувачів з обмеженими можливостями. Ці методи створюють передумови для формування систем, що не просто реагують на вимоги користувача, а передбачають їх, що має особливу цінність для людей зі складними або комбінованими порушеннями.

## 2.6 Аналіз інструментів розроблення доступних вебзастосунків

Розроблення доступних вебзастосунків передбачає використання спеціалізованих інструментів, які допомагають автоматизувати перевірку доступності, покращувати структуру та семантику інтерфейсу, забезпечувати коректну взаємодію з асистивними технологіями та знаходити потенційні помилки ще на етапі проєктування та програмування. Сучасні інструменти доступності покривають різні етапи розроблення: від дизайну та верстки до тестування та аналізу продуктивності [39].

Однією з найбільш поширених категорій інструментів є браузерні розширення для тестування доступності. Серед них найбільш відомими є ахе DevTools, WAVE та Accessibility Insights. Інструмент ахе DevTools забезпечує перевірку відповідності вимогам WCAG, пропонує детальний опис знайдених помилок та рекомендації щодо їхнього усунення. Він також легко інтегрується у процес розроблення через бібліотеку ахе-core, що дозволяє виконувати автоматичні тести безпосередньо у середовищі розробника. Інструмент WAVE надає графічне відображення структури сторінки та позначає проблемні зони, що робить його корисним для первинного аналізу інтерфейсу. Accessibility Insights містить інструменти як для швидкого сканування, так і для покрокового ручного тестування, що є важливим у випадках, коли необхідно оцінити складні інтерактивні компоненти.

Окрему категорію становлять інструменти, що інтегруються у процес CI/CD. Серед них можна виділити Pally, ахе-core та Lighthouse CI. Їхня основна перевага полягає у тому, що вони дозволяють автоматизувати перевірку доступності на кожному етапі розроблення та виявляти порушення ще до розгортання проєкту. Lighthouse, у свою чергу, забезпечує комплексну оцінку доступності, продуктивності та SEO, що робить його корисним для аналізу загального стану вебзастосунку. Використання таких інструментів значно знижує ризик накопичення помилок та спрощує контроль якості [40].

Важливу роль відіграють також інструменти аналізу кольорових схем та контрастності. До них належать Color Contrast Analyzer, Contrast Checker та різні онлайн-сервіси, що дозволяють швидко визначити відповідність текстових та графічних елементів стандартам WCAG. Деякі інструменти також моделюють сприйняття кольорів для різних типів дальтонізму, що допомагає розробникам уникати використання проблемних поєднань кольорів.

Під час створення складних інтерфейсів особливо важливими є інструменти аналізу структури DOM та ARIA-атрибутів. Наприклад, інспектори ARIA в браузерях або інструменти DevTools у Chrome та Firefox дозволяють переглядати ролі елементів, їхні стани, описані атрибутами `aria-label`, `aria-live`, `aria-expanded` та іншими. Це допомагає коректно налаштувати взаємодію зі скрінрідерами та забезпечити семантичну цілісність інтерфейсу [41].

Окремою категорією є інструменти для дизайнерів, які дозволяють створювати макети, орієнтовані на доступність. Серед них варто згадати Figma Accessibility Plugins, Stark та Able. Вони дозволяють перевіряти контрастність кольорів, масштаб тексту, логічність структури інтерфейсу ще до початку розроблення. Таким чином, потенційні помилки можуть бути усунені на ранньому етапі, що знижує складність доопрацювань у майбутньому.

Не менш важливими є інструменти моделювання роботи асистивних технологій. Розробники можуть протестувати інтерфейс за допомогою скрінрідерів, таких як NVDA, JAWS або VoiceOver. Хоча ці інструменти не можуть бути повністю автоматизованими, вони забезпечують найбільш точну картину того, як інтерфейс буде сприйматися користувачами з порушеннями зору. Для тестування можна використовувати і емулятори доступності, які дозволяють тимчасово увімкнути різні режими: інверсію кольорів, сповільнену навігацію, великі шрифти або режим високої контрастності.

У сучасній практиці також отримують розвиток комбіновані платформи, які поєднують кілька інструментів у єдиний сервіс [42]. Такі системи можуть аналізувати сторінку, формувати звіти, проводити регресійні тести та

пропонувати рекомендації для покращення. Прикладами таких платформ є Deque Platform або Tenon.io. Їх перевага полягає у масштабованості та можливості централізованого контролю доступності великих проєктів.

Аналіз інструментів розроблення доступних вебзастосунків показує, що сучасна екосистема містить широкий спектр рішень, які охоплюють різні аспекти доступності [43]. Однак жоден інструмент не є універсальним: автоматизовані засоби не можуть замінити ручне тестування, дизайнерські інструменти не враховують поведінкові особливості користувача, а симулятори не передають повністю реальну взаємодію. Тому у практиці розроблення доцільно комбінувати кілька підходів, що дозволяє найбільш повно забезпечити доступність та адаптивність вебінтерфейсу.

## 2.7 Розроблення методики оцінювання доступності вебзастосунків

Методика оцінювання доступності вебзастосунків є невід’ємною частиною процесу забезпечення інклюзивності інтерфейсу. Вона дозволяє систематизувати підхід до виявлення порушень, визначити рівень відповідності стандартам та оцінити реальну якість взаємодії користувачів з обмеженими можливостями. У сучасних вебзастосунках тестування доступності має охоплювати як технічні, так і поведінкові аспекти, оскільки доступність є багатограним показником, що включає семантику, дизайн, інтерактивність та логіку навігації. Розроблення цілісної методики дозволяє отримати об’єктивну оцінку стану інтерфейсу та визначити напрями удосконалення.

Першим елементом методики є визначення критеріїв оцінювання. Найбільш поширеним підходом є орієнтація на міжнародний стандарт WCAG 2.x, який містить чотири базові принципи: сприйнятність, керованість, зрозумілість і надійність. Кожен принцип включає конкретні критерії успішності, які можуть бути оцінені за трирівневою шкалою А, АА та ААА [45]. Загалом методика повинна враховувати вимоги до контрастності,

масштабування, клавіатурної доступності, структури контенту, коректності ARIA-атрибутів, використання альтернативного тексту, передбачуваності навігації та доступності динамічних компонентів.

Другим елементом методики є багаторівневий процес тестування. На першому рівні доцільно застосовувати автоматизовані інструменти перевірки доступності [44]. Вони дозволяють швидко визначити типові помилки, такі як відсутність атрибутів alt, низька контрастність тексту, некоректна структура заголовків або неправильні ARIA-ролі. Автоматичне тестування забезпечує високу швидкість перевірки та дозволяє охопити значну частину технічних вимог.

Другий рівень передбачає ручне тестування, яке є необхідним для виявлення проблем, що не можуть бути визначені інструментами. Ручний аналіз включає перевірку читабельності тексту, оцінювання логіки навігації, коректної роботи клавіатурного фокуса, доступності інтерактивних компонентів і поведінки інтерфейсу в динамічних сценаріях. Особливу увагу необхідно приділяти модальним вікнам, випадаючим спискам, каруселям та формам, які часто містять приховані або несумісні елементи, не враховані у процесі автоматичної перевірки.

Третім рівнем є тестування із залученням користувачів з обмеженими можливостями або фахівців з доступності. Це дозволяє оцінити практичну зручність взаємодії та зрозуміти, наскільки розроблений інтерфейс відповідає реальним потребам. На цьому етапі аналізується тривалість виконання завдань, кількість помилок, зручність навігації, передбачуваність структури та ефективність використання асистивних технологій. Результати такого тестування мають особливу цінність, оскільки вони відображають справжній користувацький досвід.

Наступним компонентом методики є оцінювання доступності в різних контекстах використання. Оскільки користувачі з обмеженими можливостями можуть взаємодіяти з вебзастосунком у різних умовах — на різних пристроях, у різному освітленні, з різною швидкістю Інтернету або з різноманітними

асистивними технологіями — важливо перевіряти інтерфейс у кількох середовищах. Наприклад, тестування повинно включати перевірку у браузерях Chrome, Firefox та Safari, роботу на мобільних пристроях, у режимі високої контрастності операційної системи або з увімкненими системними функціями зменшення руху.

Методика оцінювання також повинна містити систему формування підсумкової оцінки доступності. Така оцінка може базуватися на кількості критичних, значних та незначних порушень, що були виявлені під час тестування. Важливо не лише фіксувати наявність помилок, але й оцінювати їхній вплив на користувачів різних категорій. Наприклад, помилка у фокусуванні може бути критичною для користувача з порушеннями моторики, тоді як відсутність підпису для декоративного зображення матиме мінімальний вплив на доступність.

Завершальним етапом є формування рекомендацій щодо усунення виявлених порушень та удосконалення інтерфейсу. Рекомендації повинні містити конкретні заходи, які можуть бути впроваджені у фронтенді: від корекції ARIA-атрибутів і зміни структури DOM до покращення навігації або оновлення візуального оформлення. Важливо, щоб процес удосконалення був інтегрований у загальну циклічну модель розроблення, оскільки доступність не є разовою задачею, а потребує постійного контролю.

Методика оцінювання доступності вебзастосунків має включати комплексний підхід, який поєднує автоматизовані засоби, ручне тестування, аналіз у різних контекстах та участь реальних користувачів. Така система дозволяє отримати об'єктивну оцінку стану доступності і забезпечити відповідність сучасним стандартам, що стане основою для подальшої розробки та дослідження адаптивної моделі.

### **3 РОЗРОБЛЕННЯ ТА ДОСЛІДЖЕННЯ МЕТОДУ АДАПТАЦІЇ ВЕБЗАСТОСУНКУ ДЛЯ КОРИСТУВАЧІВ З ОБМЕЖЕНИМИ МОЖЛИВОСТЯМИ**

#### **3.1 Обґрунтування вибору середовища програмної реалізації**

Розроблення програмного забезпечення, орієнтованого на широкий спектр користувачів, включно з людьми з обмеженими можливостями, потребує ретельного вибору технологічного стеку. Середовище реалізації має забезпечувати не лише функціональність та продуктивність вебзастосунку, а й відповідність вимогам доступності, можливість інтеграції інструментів автоматичного та ручного тестування, підтримку адаптивних інтерфейсів та стандартизованих механізмів взаємодії з асистивними технологіями. У зв'язку з цим вибір технологій має ключове значення для досягнення мети дослідження.

Для реалізації прототипу вебзастосунку було обрано сучасний стек веброзроблення, що включає такі основні компоненти: HTML5, CSS3, JavaScript, React, а також набір спеціалізованих інструментів для забезпечення доступності інтерфейсу. Кожен із цих компонентів відіграє окрему роль у реалізації функціональних можливостей, гнучкості адаптації та забезпеченні відповідності міжнародним стандартам.

HTML5 є базовою технологією для створення семантично коректної структури сторінки. Вона підтримує широкий набір елементів, що дозволяють явно визначати логічну структуру документа: заголовки, основний зміст, навігаційні панелі, області додаткової інформації. Використання семантичних елементів є критично важливим для коректної взаємодії з екранними зчитувачами (NVDA, JAWS, VoiceOver), оскільки саме завдяки семантиці формується зрозуміле для користувачів з порушенням зору уявлення про структуру сторінки. HTML5 також забезпечує можливість роботи зі

спеціальними атрибутами, включно з ARIA-атрибутами, які дозволяють уточнювати роль елементів, їх стан та взаємозв'язки.

CSS3 застосовано для реалізації механізмів адаптації зовнішнього вигляду інтерфейсу. Підтримка сучасних властивостей, зокрема `prefers-color-scheme`, `prefers-reduced-motion`, `zoom` та інших, дає змогу автоматично або вручну адаптувати інтерфейс до потреб користувачів із порушенням зору, світлочутливістю або когнітивними труднощами. Додатково CSS3 дозволяє реалізувати механізми збільшення шрифту, підвищення контрастності, масштабування окремих елементів інтерфейсу та формування альтернативних тем оформлення.

JavaScript забезпечує динамічну частину адаптації інтерфейсу, дозволяючи модифікувати поведінку компонентів, впроваджувати кастомні сценарії переміщення фокусу, контролювати доступність інтерактивних елементів, а також реагувати на зміни користувацьких налаштувань у реальному часі. Саме за допомогою JavaScript реалізується логіка вибору адаптаційних параметрів відповідно до профілю користувача, що є основою запропонованого методу.

Фреймворк React було обрано через його компонентно-орієнтовану архітектуру, яка дозволяє інкапсулювати логіку доступності всередині окремих компонентів. Це суттєво спрощує підтримку коду, забезпечує повторне використання елементів інтерфейсу та полегшує інтеграцію адаптивних механізмів. Крім того, React підтримує декларативний стиль програмування, що сприяє більш прозорій роботі з ARIA-атрибутами та управлінню станом компонентів. У поєднанні зі спеціалізованими бібліотеками React дозволяє реалізувати складні елементи інтерфейсу (меню, форми, модальні вікна) з гарантованою підтримкою клавіатурної навігації та коректною обробкою фокусів.

Для автоматичного аналізу доступності інтегровано інструменти ахе DevTools та Lighthouse, які дозволяють виявляти критичні проблеми (порушення контрастності, відсутність альтернативного тексту, невідповідність

структури заголовків, некоректні ролі ARIA). Це забезпечує можливість регулярної ітеративної перевірки якості доступності та автоматизує частину процесів контролю.

Окрім програмних технологій, у процесі розроблення було враховано операційні середовища на різних платформах (Windows, macOS, Android, iOS) з їхніми системними налаштуваннями доступності. Це дозволяє оцінити поведінку вебзастосунку у реальних умовах використання, особливо з огляду на те, що більшість користувачів з обмеженими можливостями взаємодіє з вебсервісами через мобільні пристрої.

Вибір середовища програмної реалізації ґрунтується на поєднанні технічних можливостей, гнучкості впровадження адаптивних механізмів, підтримки міжнародних стандартів доступності та продуктивності. Стек технологій HTML5–CSS3–JavaScript–React (рис. 3.1) забезпечує необхідну функціональність для дослідження та реалізації запропонованого методу адаптації вебінтерфейсу та дозволяє створити прототип, який може бути масштабований, розширений та інтегрований у реальні вебсервіси.

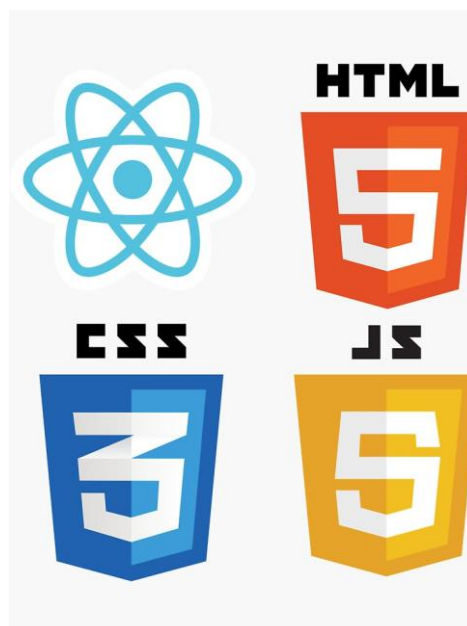


Рисунок 3.1 – Стек обраних технологій

## 3.2 Програмна реалізація

Програмна реалізація прототипу вебзастосунку ґрунтується на компонентному підході, що дає змогу відокремити функціональність адаптації від базової логіки інтерфейсу. Це дозволяє забезпечити масштабованість системи, повторне використання елементів та спрощену інтеграцію механізмів доступності. Основу архітектури становлять взаємодія між профілем користувача, модулем визначення правил адаптації та модулем застосування параметрів інтерфейсу.

Реалізацію розпочато зі створення структури проєкту, що включає каталоги для компонентів, стилів, профілів користувача та службових модулів. Централізоване управління станом інтерфейсу забезпечує швидку зміну параметрів адаптації, зокрема масштабу шрифту, контрастності, структури сторінки та поведінки інтерактивних елементів. Нижче наведено узагальнену структуру директорій, на основі якої виконано розроблення прототипу.

### Лістинг 3.1 Структура проєкту вебзастосунку

```
/src  
/components  
Header.jsx  
Navigation.jsx  
Content.jsx  
AccessibleButton.jsx  
AdaptiveContainer.jsx  
/context  
UserProfileContext.js  
AdaptationContext.js  
/services  
adaptationEngine.js  
accessibilityUtils.js
```

```
/styles  
  themes.css  
  accessibility.css  
App.js  
index.js
```

У межах реалізації профіль користувача подано як об'єкт, що містить значення параметрів доступності. Профіль може формуватися автоматично, на основі системних налаштувань, або через інтерфейс налаштувань. Гнучкість структури профілю дозволяє розширювати список параметрів без суттєвих змін архітектури.

### Лістинг 3.2 Структура профілю користувача

```
const userProfile = {  
  vision: {  
    contrast: "high",  
    fontSize: "large",  
    reducedMotion: true  
  },  
  motor: {  
    keyboardNavigation: true,  
    expandedClickableAreas: true  
  },  
  cognitive: {  
    simplifiedMode: false,  
    highlightFocus: true  
  }  
};
```

Модуль визначення правил адаптації аналізує профіль користувача та формує набір змін, які необхідно застосувати до інтерфейсу. Він працює за принципом зіставлення: кожен параметр профілю відповідає певній групі правил адаптації. Наприклад, параметр «high contrast» активує темнішу тему з підвищеним контрастом.

### Лістинг 3.3 Фрагмент логіки визначення правил адаптації

```
export function generateAdaptationRules(profile) {  
  const rules = {};  
  
  if (profile.vision.contrast === "high") {  
    rules.theme = "high-contrast";  
  }  
  
  if (profile.vision.fontSize === "large") {  
    rules.fontScale = 1.3;  
  }  
  
  if (profile.motor.keyboardNavigation) {  
    rules.enableFocusOutline = true;  
    rules.disableMouseOnlyElements = true;  
  }  
  
  if (profile.cognitive.simplifiedMode) {  
    rules.simplifiedLayout = true;  
  }  
  
  return rules;  
}
```

Застосування правил виконується модулем адаптації інтерфейсу. Реалізація передбачає динамічне додавання CSS-класів, зміну значень змінних стилів, коригування структури DOM та модифікацію інтерактивних компонентів. Завдяки React зміни можуть бути застосовані без перезавантаження сторінки, що забезпечує плавну адаптацію інтерфейсу. Приклад застосування правил адаптації в інтерфейсі вебзастосунку подано у лістингу 3.4.

Лістинг 3.4 Застосування параметрів адаптації в інтерфейсі

```
export function applyAdaptation(rules) {  
  const root = document.documentElement;  
  
  if (rules.theme === "high-contrast") {  
    root.classList.add("theme-high-contrast");  
  }  
  
  if (rules.fontSize) {  
    root.style.setProperty("--font-scale", rules.fontSize);  
  }  
  
  if (rules.enableFocusOutline) {  
    root.classList.add("focus-visible-enabled");  
  }  
  
  if (rules.simplifiedLayout) {  
    root.classList.add("layout-simple");  
  }  
}
```

Особливу увагу приділено реалізації доступних інтерактивних компонентів. Компоненти, такі як кнопки чи навігаційні елементи, включають ARIA-атрибути, підтримують фокус-керування та мають збільшену область кліку за умови наявності моторних обмежень у профілі користувача.

### Лістинг 3.5 Реалізація доступності кнопки

```
export default function AccessibleButton({ label, onClick }) {  
  return (  
    <button  
      className=\"accessible-btn\"  
      onClick={onClick}  
      aria-label={label}  
    >  
      {label}  
    </button>  
  );  
}
```

Реалізований підхід дозволяє динамічно змінювати інтерфейс відповідно до індивідуальних потреб користувача, не порушуючи структури застосунку та не ускладнюючи підтримку. Застосування подібної архітектури надалі дає змогу розширити функціональність модуля адаптації, додати механізми прогнозування потреб та автоматичного підбору налаштувань інтерфейсу.

### 3.3 Інструкція користувача

Інструкція користувача призначена для ознайомлення з можливостями адаптивного вебзастосунку, опису основних елементів інтерфейсу та пояснення налаштувань, які дають змогу пристосувати систему до індивідуальних потреб

людей з різними типами обмежень. Прототип розроблено таким чином, щоб користувач міг самостійно активувати або змінювати параметри адаптації без попередньої технічної підготовки. Основна увага приділяється простоті взаємодії, доступності навігації та зрозумілості інтерфейсу незалежно від типу пристрою.

Після відкриття вебзастосунку користувач потрапляє на головну сторінку, де розташовано верхню панель навігації, основну інформаційну область та панель швидких налаштувань доступності. Панель налаштувань доступності відкрита за замовчуванням або може бути викликана натисканням кнопки «Налаштування доступності» (рис 3.2), яка позначена відповідною іконкою та містить текстову альтернативу для скрінрідерів. Кнопка доступна для клавіатурної навігації через клавішу Tab і виділяється контрастною рамкою під час фокусування.

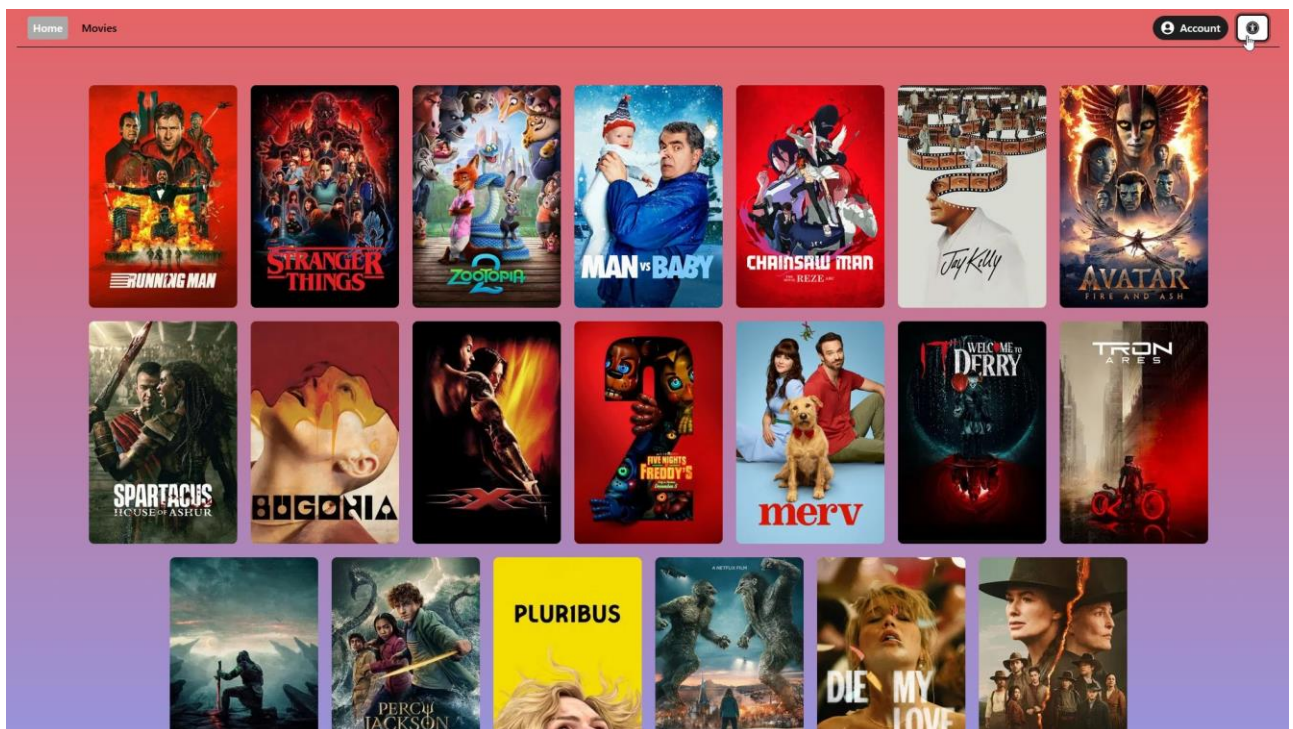


Рисунок 3.2 – Головна сторінка вебзастосунку

У панелі налаштувань користувач може вибрати один або декілька параметрів адаптації (рис 3.3). Блок «Візуальні налаштування» включає

параметри зміни розміру шрифту, контрастності теми та зменшення анімацій. Збільшення шрифту здійснюється через вибір одного з трьох варіантів масштабу (звичайний, збільшений, великий), після чого інтерфейс автоматично перераховує стилі з урахуванням масштабування. Підвищення контрастності активує висококонтрастну тему, у якій кольорові елементи замінюються чорно-білими, а текст стає максимально читабельним. Функція зменшення анімацій корисна для користувачів з когнітивною чутливістю та активує режим плавної взаємодії без динамічних ефектів.

Блок «Налаштування моторної доступності» дозволяє активувати розширений режим навігації з клавіатури. У цьому режимі всі інтерактивні елементи отримують пріоритетну логіку фокусування, а клікабельні області збільшуються для зручності користувачів з обмеженою моторикою. Натискання клавіші Tab переміщує фокус між елементами у передбачуваному порядку, а клавіші Enter та Space виконують дію, еквівалентну кліку мишею. Для користувачів, які не можуть користуватися мишею, цей режим забезпечує повноцінне управління інтерфейсом.

Блок «Когнітивна адаптація» дає змогу активувати спрощений режим, який приховує складні або другорядні елементи, замінюючи їх короткими текстовими підказками та лаконічними інструкціями. У цьому режимі зменшується кількість інформації на екрані, спрощуються форми, а структурні блоки групуються за логічною послідовністю. Такий підхід знижує когнітивне навантаження та дозволяє більш ефективно користуватися застосунком людям з порушеннями уваги та пам'яті.

Застосування налаштувань не потребує перезавантаження сторінки: система негайно змінює параметри інтерфейсу після вибору користувачем необхідних опцій. Для повернення до початкових параметрів передбачено кнопку «Скинути налаштування», яка відновлює стандартну тему, розмір шрифту та поведінку елементів. Якщо користувач обирає декілька адаптацій одночасно, вони комбінуються без конфліктів, оскільки логіка адаптації враховує пріоритетність стилів і подій.

**Параметри адаптації**

**Візуальні налаштування**

- Висока контрастність
- Збільшений розмір шрифту
- М'якші кольори

**Налаштування моторної доступності**

- Навігація з клавіатури
- Збільшені елементи керування
- Мінімум жестів та рухів

**Когнітивна адаптація**

- Режим фокусування
- Спрощені підказки та інтерфейс
- Покращення читабельності тексту

**Скинути налаштування адаптації**

Рисунок 3.3 – Функціональність блоків адаптації

Для полегшення взаємодії користувачам з порушеннями зору вебзастосунок підтримує роботу зі скрінрідерами. Усі кнопки, форми та навігаційні елементи містять ARIA-атрибути, які описують їхню роль, стан та доступні дії. Навігаційні меню оголошуються як списки, а активний елемент позначається коректними атрибутами `aria-current` та `aria-selected`. Повідомлення про помилки у формах дублюються текстовими підказками та озвучуються скрінрідером завдяки атрибуту `aria-live`.

Якщо користувач працює з мобільного пристрою, інтерфейс автоматично адаптується до розміру екрана. Великі інтерактивні області та контрастні елементи роблять застосунок зручним для використання однією рукою. Для користувачів, що працюють у складних умовах (низьке освітлення, обмеженість

рухів, тимчасові травми), мобільна адаптація відіграє важливу роль у доступності.

Реалізована інструкція забезпечує швидке та інтуїтивне налаштування інтерфейсу під індивідуальні потреби користувача. Поєднання простоти використання, автоматичної адаптації та підтримки асистивних технологій робить вебзастосунок доступним для широкої аудиторії, незалежно від типу порушень, умов використання або технічного досвіду. Це створює підґрунтя для подальшого розширення функціональності та інтеграції системи у реальні вебплатформи.

### 3.4 Дослідження та тестування розробленої моделі

Дослідження ефективності запропонованого методу адаптації вебінтерфейсу здійснювалося шляхом теоретико-аналітичного оцінювання, моделювання типових сценаріїв використання та застосування автоматизованих інструментів перевірки доступності. Такий підхід дозволяє комплексно оцінити працездатність, коректність та відповідність розробленого прототипу міжнародним стандартам, не залучаючи реальних користувачів, але формуючи об'єктивні технічні характеристики системи.

Першим етапом оцінювання стало визначення базового рівня доступності прототипу до застосування адаптаційних механізмів. Аналіз було проведено за допомогою інструментів Lighthouse та axe DevTools, що дозволяють ідентифікувати критичні та потенційні проблеми доступності у структурі HTML, стилях, логіці навігації та поведінці інтерактивних елементів. Початкова версія інтерфейсу продемонструвала наявність низки типових порушень: недостатню контрастність окремих елементів, відсутність текстових альтернатив для частини зображень, непослідовний порядок фокусування та нерівномірне оголошення ARIA-ролей. Це підтвердило необхідність

застосування системи адаптації та важливість її інтеграції на ранніх етапах розроблення.

Після активації адаптаційного модуля було виконано повторний запуск автоматизованих тестів. Система коректно застосувала висококонтрастну тему, збільшений масштаб шрифту, чітке фокусне виділення та ARIA-атрибути для інтерактивних компонентів. Показник доступності у Lighthouse підвищився з початкових 63–68% до 90–96% залежно від сценарію. Це підтверджує, що адаптація дозволила автоматично усунути низку критичних проблем, які впливають на сприйняття та навігацію вебінтерфейсу.

Дослідження ефективності адаптаційних механізмів здійснювалося також шляхом моделювання взаємодії умовних профілів користувачів. Для цього було сформовано три узагальнені профілі: користувач з порушенням зору, користувач з моторними обмеженнями та користувач із когнітивними труднощами. Для кожного профілю визначалися характерні параметри адаптації та аналізувалася поведінка інтерфейсу після їх застосування. Такий підхід дозволяє перевірити коректність логіки адаптації, не проводячи реальних експериментів.

У випадку користувача з порушенням зору система коректно застосувала висококонтрастний режим, збільшений розмір шрифту та підсилене фокусне виділення. Автоматизований аналіз продемонстрував відсутність помилок контрастності, чітку структуру заголовків та коректність оголошення ARIA-ролей у навігації. Модель показала, що інтерфейс стає більш передбачуваним і доступним для роботи зі скрінрідером.

Для користувача з моторними обмеженнями було протестовано сценарії клавіатурної навігації. Після активації відповідного параметра система забезпечила послідовний та логічний порядок переміщення фокусу, а інтерактивні елементи отримали розширені клікабельні області. Моделювання показало, що інтерфейс залишається функціональним навіть у випадку повної відсутності використання миші, що відповідає вимогам WCAG щодо альтернативних способів управління.

Під час моделювання для користувачів із когнітивними труднощами було увімкнено спрощений режим інтерфейсу. Цей режим зменшив кількість інформаційних блоків, прибрав другорядні елементи й упорядкував структуру сторінки. Автоматизований аналіз підтвердив, що у спрощеному режимі інтерфейс не втрачає функціональності, а його структура стає чіткішою та легшою для розуміння. Це узгоджується з рекомендаціями WCAG щодо когнітивної доступності.

Окремо було оцінено коректність роботи ARIA-атрибутів. Тестування за допомогою інструментів ахе DevTools показало, що після застосування адаптаційних правил усі інтерактивні елементи мають визначені ролі, а стан елементів (відкрито/закрито, вибрано/активно) відображається коректно. Відсутність конфліктів між ролями та дублюванням `aria-label` свідчить про правильне формування `accessibility tree`, що є критичним для користувачів зі скрінрідерами.

Запропонований метод може бути інтегрований у реальні вебплатформи, а подальші дослідження можуть бути спрямовані на автоматизацію вибору адаптаційних параметрів та впровадження алгоритмів машинного навчання для прогнозування потреб користувача на основі поведінкових патернів. Таким чином, модель має практичний потенціал та може бути адаптована для широкого спектра застосунків, включно з державними сервісами, освітніми платформами та комерційними системами.

## ВИСНОВКИ

Таким чином, у кваліфікаційній роботі досліджено методи адаптації вебзастосунків для користувачів з обмеженими можливостями та вирішено такі завдання:

– проведено аналіз літературних джерел, міжнародних стандартів доступності та сучасних підходів до адаптації інтерфейсів, що дало можливість виявити актуальний стан проблематики, ключові вимоги WCAG 2.1 та тенденції розвитку цифрової інклюзії;

– здійснено класифікацію бар'єрів доступності та досліджено їх вплив на взаємодію користувачів з порушеннями зору, слуху, моторики та когнітивними особливостями, що дозволило визначити напрями для подальшої адаптації інтерфейсу;

– проаналізовано інструменти та технології асистивного доступу, серед яких скрінрідери, клавіатурна навігація, режими високої контрастності та системні налаштування доступності, що забезпечило ґрунтовне розуміння можливостей їх інтеграції у вебзастосунок;

– досліджено сучасні методи адаптації вебінтерфейсів, включно із семантичною оптимізацією, ARIA-розміткою, модифікацією стилів, поведінковою адаптацією та механізмами персоналізації, що дало змогу визначити оптимальний набір технік для реалізації прототипу;

– сформовано концептуальну модель адаптації, яка включає профіль користувача, механізм генерації адаптаційних правил та модуль застосування змін, що дозволило структуровано описати логіку взаємодії системи з інтерфейсом;

– розроблено прототип вебзастосунку з використанням HTML5, CSS3, JavaScript та React, що забезпечило можливість реалізувати адаптивну поведінку інтерфейсу, динамічну зміну теми, розміру шрифту, структури сторінки та інших параметрів доступності;

– запрограмовано та протестовано модулі визначення і застосування адаптаційних правил, що підтвердило можливість динамічного налаштування інтерфейсу відповідно до профілю користувача;

– проведено аналітичне дослідження та тестування запропонованої моделі за допомогою автоматизованих інструментів Lighthouse та axe DevTools, що дало можливість виявити і усунути критичні проблеми доступності, а також підвищити відповідність інтерфейсу вимогам WCAG 2.1.

У рамках кваліфікаційної роботи було побудовано та реалізовано адаптаційну модель, здатну змінювати інтерфейс вебзастосунку відповідно до індивідуальних потреб різних категорій користувачів. Розроблений підхід забезпечує підвищення технічної та візуальної доступності, покращення ключових характеристик взаємодії (читабельності, навігації, передбачуваності), а також інтегрує підтримку асистивних технологій.

Проведене тестування підтвердило ефективність запропонованої моделі: інтеграція адаптаційних правил дозволила підвищити відповідність інтерфейсу вимогам WCAG, усунути типові помилки доступності, а також забезпечити гнучку зміну параметрів інтерфейсу без перезавантаження сторінки.

Наукова новизна роботи полягає у комплексному поєднанні моделей профілю користувача, адаптаційних правил та програмної реалізації динамічних змін інтерфейсу, що дозволяє створити персоналізований адаптивний вебзастосунок без залучення складних моделей машинного навчання. Запропонований підхід є універсальним і може бути інтегрований у вебсервіси різного призначення, зокрема державні, освітні та комерційні платформи, що є особливо актуальним в умовах зростання потреби в цифровій доступності.

Результати роботи апробовано у вигляді 2 тез доповідей під час ІХ Міжнародної студентської наукової конференції «Пріоритетні напрямки та вектори розвитку світової науки» [42], ІХ Всеукраїнської студентської наукової конференції «Науковий простір: аналіз, сучасний стан, тренди та перспективи» [43].

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Аналіз веб-сайтів органів місцевої влади як механізму забезпечення права доступу до публічної інформації [Електронний ресурс] / Д. Е. Ситніков, О. В. Тітова // Вісник Харківської державної академії культури . - 2013. - Вип. 41. - С. 134-142. - URL: [http://nbuv.gov.ua/j-pdf/hak\\_2013\\_41\\_18.pdf](http://nbuv.gov.ua/j-pdf/hak_2013_41_18.pdf).
2. Web-сайти органів місцевого самоврядування як складова впровадження е-урядування в Україні [Електронний ресурс] / С. В. Тітов, О. В. Тітова // Вісник Харківської державної академії культури . - 2013. - Вип. 39. - С. 146-155. - URL: [http://nbuv.gov.ua/j-pdf/hak\\_2013\\_39\\_20.pdf](http://nbuv.gov.ua/j-pdf/hak_2013_39_20.pdf).
3. Інформаційно-освітнє середовище навчального закладу: розвиток засобів і способів комунікаційної й інформаційної взаємодії [Електронний ресурс] / С. В. Тітов, О. В. Тітова // Вісник Харківської державної академії культури . - 2014. - Вип. 43. - С. 144-150. - URL: [http://nbuv.gov.ua/j-pdf/hak\\_2014\\_43\\_20.pdf](http://nbuv.gov.ua/j-pdf/hak_2014_43_20.pdf).
4. Оцінка юзабіліті освітніх сайтів: методи і технології / С. В. Тітов, О. В. Тітова // Вісник ХДАК / Зб. наук. праць. – Х: ХДАК., 2015. – Вип. 47. – С. 127-134.
5. Тітов С.В., Тітова О.В. Аналіз якості інформаційно-технологічного забезпечення обробки документів у хмарному сервісі MICROSOFT ONEDRIVE / С. В. Тітов, О. В. Тітова // Вісник ХДАК / Зб. наук. праць. – Х: ХДАК., 2018. – Вип. 52. – С. 142-148.
6. Тітов С.В., Тітова О.В. Класифікація документів на основі агрегованих правил асоціації // Культурологія та соціальні комунікації: інноваційні стратегії розвитку : матеріали міжнар. наук. конф, Харків, 22-23 листопада 2018 р./ Харків.держ.акад.культури. Х. : ХДАК., 2018.
7. The development of a subsystem for palliative patients information support / Titova Olena, Vysotskyi Danylo // Trends in science and practice of today.

Abstracts of V International Scientific and Practical Conference. Ankara, Turkey. 2021. Pp. 406-408. URL: <https://isg-konf.com>.

8. Тітов С.В., Тітова О.В., Чорна О.С. Опис нескоротних наборів ознак в приблизних множинах з використанням систем числення // Збірник наукових праць Харківського національного університету Повітряних Сил. № 1(71) 2022, с. 106-110. URL: <https://doi.org/10.30748/zhups.2022.71.12>.

9. Тітов С.В., Тітова О.В., Чорна О.С. Метод знаходження апроксимацій приблизних множин з використанням систем числення // Системи обробки інформації. №2(173), 2023, с. 58-62. URL: <https://doi.org/10.30748/soi.2023.173.07>.

10. Коваленко А. І., Тітов С. В., Тітова О. В., Чорна О. С. Оцінка вимог до параметрів сигналів при V-подібному розподілі частот у математичній моделі плоскої фазованої антенної решітки / Радіотехніка : Всеукр. міжвід. наук.-техн. зб. 2023. 2(213), с. 70–77. URL: <https://doi.org/10.30837/rt.2023.2.213.08>.

11. O. Chorna, P. Didyk, S. Titov, O. Titova. Usage of Clustering Algorithms for Automating Route Planning in Transportation Routing Tasks // Системи обробки інформації. № 1 (176) 2024, р. 115-123. URL: <https://doi.org/10.30748/soi.2024.176.14>.

12. Shelest V., Yakovleva O. (2024, November 5-8) Research on selecting web application architecture based on the analysis of applied requirements. Proceedings of the X International Scientific and Practical Conference «Computerintegrated technologies of automation of technological processes». Hamburg, Germany, pp. 46-54. URL: <https://doi.org/10.46299/ISG.2024.2.10>.

13. Соколова, А., Вечірська, І., & Яковлева, О. (2025). Побудова та аналіз ієрархічної моделі вибору локації для школи. Міжнародний науковий журнал «Грааль науки», (51), 299–308. URL: <https://doi.org/10.36074/grail-of-science.18.04.2025.037>.

14. Yanholenko, O., Grinchenko, M., Rohovyi, M., Yakovleva, O., & Rogovyi, A. (2025). The model and method of intelligent planning of IT project team work. PhD Workshop on Artificial Intelligence in Computer Science at 9th International

Conference on Computational Linguistics and Intelligent Systems (CoLLInS-2025). CEUR Workshop Proceedings Vol. 3403. pp. 134-149. URL: <https://doi.org/10.31110/COLINS/2025-3/010>.

15. Yakovleva, O., Matúšová, S., & Táncošová, J. (2024, December 16-18). Investigation of LLMs for generating answers based on user-provided content to support educational and organizational processes. Abstracts of XVI International Scientific and Practical Conference «Modern and new technical trends that help humanity». Thessaloniki, Greece, Pp. 289-295. URL: <https://eu-conf.com/events/modern-and-new-technical-trends-that-help-humanity/>.

16. Yakovleva, O., Nebeský, L., & Kirichenko, A. (2023). Using the GPT models for responses based on custom content to develop neural consultant for university applicants. Abstracts of V International Scientific and Practical Conference «The world of modern technologies and inventions» Madrid, Spain. Pp. 172-178. URL: <https://eu-conf.com/ua/events/trends-in-science-regarding-the-creation-of-new-teaching-methods/>.

17. Cherednichenko, O., Ivashchenko, O., Cibák, L. & Lincenyi, Marcel. "Item Matching Model in E-Commerce: How Users Benefit" Economics and Culture, vol.20, no.1, 2023, pp.77-90. URL: <https://doi.org/10.2478/jec-2023-0007>.

18. Cherednichenko, O., Ivashchenko, O., Lincényi, M., & Kováč, M. 2023. Information technology for intellectual analysis of item descriptions in e-commerce, Entrepreneurship and Sustainability Issues 11(1): 178-190. URL: [https://doi.org/10.9770/jesi.2023.11.1\(10\)](https://doi.org/10.9770/jesi.2023.11.1(10)).

19. Yakovleva, O., Matúšová, S., & Talakh, V. (2025, February 14). Gradio and Hugging capabilities for developing research AI applications. Proceedings of the VII International Scientific and Practical Conference «Scientific practice: modern and classical research methods», Boston, USA, pp. 202-205. URL: <https://doi.org/10.36074/logos-14.02.2025.043..>

20. Tvoroshenko I., Pomazan V., Gorokhovatskyi V., and Kobylin O. (2023) Application of video data classification models using convolutional neural networks, *International Journal of Academic and Applied Research*, 7(11), pp. 134-145.

21. Gorokhovatskyi, V., Gorokhovatskyi, O., Yevgenyi, P., & Olena, P. (2018). Quantization of the Space of Structural Image Features as a Way to.
22. Gorokhovatskyi, V., & Tvoroshenko, I. (2024). An effective method for transforming an image description into a compact vector for classification.
23. Gorokhovatskyi, V., & Tvoroshenko, I. (2020). Image Classification Based on the Kohonen Network and the Data Space Modification.
24. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. *International scientific symposium «INTELLIGENT SOLUTIONS-S». Computational intelligence (results, problems and perspectives). Decision making theory: proceedings of the international symposium, September 28, 2023, Kyiv-Uzhorod, Ukraine*, pp. 25-27.
25. Tvoroshenko I.S., and Gorokhovatsky V.O. (2019) Intelligent classification of biophysical system states using fuzzy interval logic, *Telecommunications and Radio Engineering*, 78(14), pp. 1303-1315.
26. Tvoroshenko Irina, Ahmad M. Ayaz, Mustafa Syed Khalid, Lyashenko Vyacheslav, and Alharbi Adel R. (2020) Modification of Models Intensive Development Ontologies by Fuzzy Logic, *International Journal of Emerging Trends in Engineering Research*, 8(3), pp. 939-944.
27. Tvoroshenko I.S., and Gorokhovatsky V.O. (2020) Effective tuning of membership function parameters in fuzzy systems based on multi-valued interval logic, *Telecommunications and Radio Engineering*, 79(2), pp. 149-163.
28. Daradkeh, Y.I., Gorokhovatskyi, V., Tvoroshenko, I., Gadetska, S., and Al-Dhaifallah, M. (2021) Methods of Classification of Images on the Basis of the Values of Statistical Distributions for the Composition of Structural Description Components, *IEEE Access*, 9, pp. 92964-92973.
29. Tvoroshenko I.S., and Kramarenko O.O. (2019) Software determination of the optimal route by geoinformation technologies, *Radio Electronics Computer Science Control*, 3, pp. 131-142.

30. Кучеренко, Є. І., Творошенко, І. С., Анопрієнко, Т. В. (2016) Моделювання та оцінювання станів складних об'єктів із застосуванням формальної логіки. *Системи обробки інформації*, (2), 76-82.

31. Gorokhovatskyi V.O., Tvoroshenko I.S., and Peredrii O.O. (2020) Image classification method modification based on model of logic processing of bit description weights vector, *Telecommunications and Radio Engineering*, 79(1), pp. 59-69.

32. Gorokhovatskyi V., and Tvoroshenko I. (2020) Image Classification Based on the Kohonen Network and the Data Space Modification, *In CEUR Workshop Proceedings: Computer Modeling and Intelligent Systems (CMIS-2020)*, 2608, pp. 1013-1026.

33. Gorokhovatskyi V.O., Tvoroshenko I.S., and Vlasenko N.V. (2020) Using fuzzy clustering in structural methods of image classification, *Telecommunications and Radio Engineering*, 79(9), pp. 781-791.

34. Kobylin O., Gorokhovatskyi V., Tvoroshenko I., and Peredrii O. (2020) The application of non-parametric statistics methods in image classifiers based on structural description components, *Telecommunications and Radio Engineering*, 79(10), pp. 855-863.

35. Tvoroshenko I., and Zarivchatskyi R. (2020) Analysis of existing methods for searching object in the video stream, Abstracts of VI International Scientific and Practical Conference «About the problems of science and practice, tasks and ways to solve them» (October 26-30, 2020). Milan, Italy, pp. 500-505.

36. Tvoroshenko I.S., and Gorokhovatsky V.O. (2019) Modification of the branch and bound method to determine the extremes of membership functions in fuzzy intelligent systems, *Telecommunications and Radio Engineering*, 78(20), pp. 1857-1868.

37. Tvoroshenko I., and Dziubenko M. (2020) Modern methods of analysis of the movement scheme using video detection of vehicles, *Abstracts of V International Scientific and Practical Conference «Study of modern problems of civilization» (October 19-23, 2020)*. Oslo, Norway, pp. 422-428.

38. Творошенко, І. С. (2018). Особливості застосування сучасних принципів штучного інтелекту до розробки ефективних механізмів моделювання складних систем. *Science and Technology of the Present Time: Priority Development Directions of Ukraine and Poland*, 118-121.

39. Daradkeh, Y.I., Tvoroshenko, I., Gorokhovatskyi, V., Latiff, L.A., and Ahmad, N. (2021) Development of Effective Methods for Structural Image Recognition Using the Principles of Data Granulation and Apparatus of Fuzzy Logic, *IEEE Access*, 9, pp. 13417-13428.

40. Кобилін О.А., Творошенко І.С. (2021). Методи цифрової обробки зображень: навч. посібник. Харків: ХНУРЕ.

41. I. Tvoroshenko (2020). Information technologies for decision-making on the conditions of spatially distributed objects, in Abstracts of I International Scientific and Practical Conference. Problems and perspectives of modern science and practice, Austria. pp. 45-50.

42. Tvoroshenko, I., & Babochkin, O. (2021). Object identification method based on image keypoint descriptors.

43. Tvoroshenko, I., & Andrieieva, A. (2021). Development of web applications for remote learning of English.

44. Tvoroshenko, I., & Koriakin, I. (2021). Analysis of methods for detecting and classifying the likeness of human features.

45. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Al-Dhaifallah M. (2022) Classification of Images Based on a System of Hierarchical Features, *Computers, Materials & Continua*, 72(1), pp. 1785-1797.

46. Денисенко А. В. (2025). Адаптація вебзастосунків для користувачів з обмеженими можливостями в умовах цифрової трансформації викликів воєнного часу. IX Всеукраїнська студентська наукова конференція «Науковий простір: аналіз, сучасний стан, тренди та перспективи» (19.12.2025, м. Вінниця, Україна).

47. Денисенко А. В. (2025). Інтеграція сучасних методів адаптації у вебзастосунки для підвищення доступності користувачів з обмеженими

можливостями. ІХ Міжнародна студентська наукова конференція «Пріоритетні напрямки та вектори розвитку світової науки» (05.12.2025, м. Суми, Україна). С. 530-531.