

**МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ**

**ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І
СИСТЕМ**

КВАЛІФІКАЦІЙНА РОБОТА

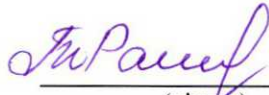
на тему: Адаптація сайту під мобільний пристрій

Здобувача вищої освіти першого (бакалаврського) рівня,
IV курсу, групи ЦДБ-21-1, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Технології цифрового дизайну»


(підпис)

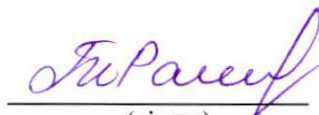
Левицького Володислава
Олександровича
(прізвище, ім'я, по батькові)

Науковий керівник


(підпис)

Ратушняк Тетяна Володимирівна,
к.ф.-м.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь,
вчене звання)

Керівник освітньо-
професійної програми


(підпис)

Ратушняк Тетяна Володимирівна,
к.ф.-м.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь,
вчене звання)

Допущено до захисту за рішенням кафедри комп'ютерних та інформаційних
технологій і систем протокол від «13» травня 2025 № 18

Завідувач кафедри


(підпис)

Філоненко Михайло
Миколайович, к.ф.-м.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь,
вчене звання)

Ірпінь – 2025

Реєстрація № 75

29.05.2025

(дата)



(підпис відповідальної особи)

Результати перевірки

до захисту
(до захисту, на доопрацювання)

29.05.2025

(дата)



(підпис)

Результат захисту

4 (добре)
(оцінка)

10.06.2025

(дата)



(підпис)

Голова комісії:



(підпис)

В.М. Франчук

Члени комісії:



(підпис)

М.М. Філоненко



(підпис)

Т.В. Ратушняк



(підпис)

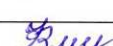
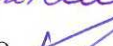
В.О. Ніжегородцев

БЛАНК ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії, секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	від 0 до 30	25		Ратушняк Т.В. 
2. Дотримання вимог щодо оформлення	від 0 до 10	8		Франчук В.М. 
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	4		Філоненко М.М.  Ратушняк Т.В. 
4. Рецензент	від 0 до 5	5		Ніжегородцев В.О.  Суховерхівка Л.П. 
РАЗОМ БАЛІВ	від 0 до 50	42		

Оцінювання кваліфікаційної роботи науковим керівником — 43 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підпис)
85	B	4 (добре)	10.06.2025	Франчук В.М.  Філоненко М.М.  Ратушняк Т.В.  Ніжегородцев В.О.  Суховерхівка Л.П. 

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	
60-69	D	
50-59	E	3 (задовільно)
35-49	FX	2 (незадовільно)

Факультет фінансів та цифрових технологій
Кафедра комп'ютерних та інформаційних технологій і систем
Галузь знань: 12 «Інформаційні технології»
Спеціальність: 122 «Комп'ютерні науки»
Освітня програма: «Технології цифрового дизайну»

Затверджую
Завідувач кафедри  Філоненко М.М.
«14» листопада 2024

Завдання на кваліфікаційну роботу
здобувачу вищої освіти
Левицькому Володиславу Олександровичу

1. Тема кваліфікаційної роботи: Адаптація сайту під мобільний пристрій
Затверджена наказом ректора від «13» листопада 2024 № 1717
2. Строк здачі закінченої роботи «12» травня 2025
3. Мета роботи: розробити та впровадити респонсивний інтерфейс для вебсайту TurboVive із забезпеченням коректного відображення і зручної взаємодії на чотирьох ключових брейк-поінтах (≤ 480 px, 481–768 px, 769–1199 px, ≥ 1200 px).
Об'єкт дослідження: інформаційні технології розробки та адаптації вебсайту під мобільний пристрій.
Предмет дослідження: методи та інструменти респонсивного вебдизайну.
Орієнтовна база дослідження: КП «Микуличі».
4. Вихідні дані до роботи: технічне завдання замовника; перелік ключових брейк-поінтів; наукові публікації у сфері цифрового проєктування та дизайну; зразки вебсайтів, які мають адаптовані версії для різних типів смартфонів.
5. Перелік графічного матеріалу: скріншоти мобільних версій сайтів McDonalds і Adidas.
6. Зміст кваліфікаційної роботи:
ВСТУП;
РОЗДІЛ 1. Теоретико-методологічні основи адаптації сайтів під мобільні пристрої
1.1. Актуальність адаптації сайтів у сучасних умовах; 1.2. Еволюція користування інтернетом: перехід на мобільні пристрої; 1.3. Типи мобільних пристроїв і роздільні здатності екранів; 1.4. Поняття адаптивного та респонсивного дизайну; 1.5. Основні принципи UX/UI для мобільних інтерфейсів; 1.6. Огляд сучасних технологій, методів та фреймворків адаптації; 1.7. Порівняльний аналіз успішних мобільних адаптацій
РОЗДІЛ 2. Системно-аналітичне дослідження технічних рішень адаптації
2.1. Роль CSS3 та media-запитів у формуванні адаптивності; 2.2. Техніки адаптації зображень, шрифтів, блоків і навігації; 2.3. Роль метатегів і типографіки для мобільних сайтів; 2.4. Аналіз стандартів Google: Mobile-Friendly Test, Lighthouse, PageSpeed
РОЗДІЛ 3. Проєктно-прикладна частина: розробка адаптивного вебсайту
3.1. Постановка задачі та вимоги до сайту; 3.2. Проєктування структури та дизайну мобільного інтерфейсу; 3.3. Вибір середовища розробки та технологій; 3.4. Реалізація адаптивного макету: HTML, CSS, JS; 3.5. Тестування сайту на різних екранах і пристроях; 3.6. Перевірка швидкодії та юзабіліті: результати Lighthouse; 3.7. Пропозиції щодо вдосконалення розробленого рішення
ВИСНОВКИ; СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ; ДОДАТКИ

7. Календарний план виконання роботи

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	Фактично
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	До 30.10.2024	До 30.10.2024
2	I розділ (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	25.12.2024	25.12.2024
3	II розділ (аналіз експериментальної частини про методики досліджень)	15.02.2025	15.02.2025
4	III розділ (проведені теоретичні і (або) експериментальні дослідження та проєктні рішення)	14.04.2025	14.04.2025
5	Висновки	01.05.2025	01.05.2025
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	09.05.2025	09.05.2025
7	Отримання відгуку керівника	03.06.2025	03.06.2025
8	Отримання зовнішньої рецензії	05.06.2025	05.06.2025

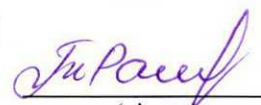
8. Дата видачі завдання «14» листопада 2024

9. Керівник кваліфікаційної роботи


(підпис)

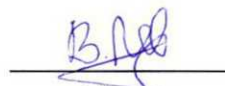
Ратушняк Т.В.
(прізвище, ініціали)

10. Керівник освітньо-професійної програми


(підпис)

Ратушняк Т.В.
(прізвище, ініціали)

11. Завдання до виконання отримав здобувач вищої освіти


(підпис)

Левицький В.О.
(прізвище, ініціали)

Анотація

Адаптація сайту під мобільний пристрій

Мета роботи: розробити та впровадити респонсивний інтерфейс для вебсайту TurboVive із забезпеченням коректного відображення і зручної взаємодії на чотирьох ключових брейк-поінтах (≤ 480 px, 481–768 px, 769–1199 px, ≥ 1200 px).

Завдання дослідження: аналіз теоретичних основ адаптивного дизайну та інструментів для його реалізації; розробка дизайн-макету мобільної версії вебсайту; налаштування адаптивності вебсайту для різних типів смартфонів; верстка вебсайту; проведення тестування розробки.

Об'єкт дослідження: інформаційні технології розробки та адаптації вебсайту під мобільний пристрій.

Предмет дослідження: методи та інструменти респонсивного вебдизайну; особливості налаштування адаптивності для мобільних версій вебсайтів.

Методи дослідження: аналіз літературних джерел – для вивчення теоретичних основ адаптивного дизайну та інструментів для його розробки, порівняльний аналіз – для оцінки функціональних можливостей різних інструментів для створення адаптивного дизайну, практичний метод – для розробки дизайн-макету та налаштування адаптивності вебсайту, метод тестування – для перевірки ефективності розробленого адаптивного вебдизайну на різних пристроях.

База дослідження: КП «Микуличі».

Одержані результати можуть бути використані у проєктуванні адаптивних вебдизайнів із застосуванням сучасних інформаційних технологій та налаштуванні адаптивності вебсайтів. засобами Figma. Прийоми та рекомендації, запропоновані у роботі, будуть корисними для вдосконалення адаптивного дизайну та покращення взаємодії користувачів з мобільними версіями вебсайтів.

Кваліфікаційна робота містить 55 сторінок, 1 таблицю, 1 рисунок, список використаних джерел з 60 найменувань, 8 додатків.

Ключові слова: цифрове проєктування, вебдизайн, адаптивний дизайн, респонсивний дизайн, UX/UI дизайн, інклюзивний вебдизайн, mobile-first, HTML, CSS, JavaScript, CSS3 media-запити, SEO-оптимізація, Flexbox, CSS Grid.

Рік виконання кваліфікаційної роботи 2025.

Рік захисту роботи 2025.

Abstract

Site adaptation for mobile devices

Purpose of the work: to develop and implement a responsive interface for the TurboVive website, ensuring correct display and convenient interaction at four key breakpoints (≤ 480 px, 481–768 px, 769–1199 px, ≥ 1200 px).

Research objectives: analysis of the theoretical foundations of adaptive design and tools for its implementation; development of a design layout for the mobile version of the website; setting up the adaptability of the website for different types of smartphones; website layout; testing the development.

Object of the research: information technologies for developing and adapting a website for a mobile device.

Subject of the research: methods and tools of responsive web design; features of setting up adaptability for mobile versions of websites.

Research methods: analysis of literary sources - to study the theoretical foundations of adaptive design and tools for its development, comparative analysis - to assess the functionality of various tools for creating adaptive design, practical method - to develop a design layout and configure the adaptability of the website, testing method - to verify the effectiveness of the developed adaptive web design on different devices.

Research base: KP "Mykulychi".

The results obtained can be used in designing adaptive web designs using modern information technologies and configuring the adaptability of websites. using Figma tools. The techniques and recommendations proposed in the work will be useful for improving adaptive design and improving user interaction with mobile versions of websites.

The qualification work contains 55 pages, 1 table, 1 figure, a list of used sources of 60 items, 8 appendices.

Keywords: digital design, web design, adaptive design, responsive design, UX/UI design, inclusive web design, mobile-first, HTML, CSS, JavaScript, CSS3 media queries, SEO optimization, Flexbox, CSS Grid.

Year of completion of the qualification work 2025.

Year of defense of the work 2025.

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ОСНОВИ АДАПТАЦІЇ САЙТІВ ПІД МОБІЛЬНІ ПРИСТРОЇ	5
1.1. Актуальність адаптації сайтів у сучасних умовах	5
1.2. Еволюція користування інтернетом: перехід на мобільні пристрої	6
1.3. Типи мобільних пристроїв і роздільні здатності екранів	8
1.4. Поняття адаптивного та респонсивного дизайну	14
1.5. Основні принципи UX/UI для мобільних інтерфейсів	15
1.6. Огляд сучасних технологій, методів та фреймворків адаптації	17
1.7. Порівняльний аналіз успішних мобільних адаптацій	19
РОЗДІЛ 2. СИСТЕМНО-АНАЛІТИЧНЕ ДОСЛІДЖЕННЯ ТЕХНІЧНИХ РІШЕНЬ АДАПТАЦІЇ.....	23
2.1. Роль CSS3 та media-запитів у формуванні адаптивності	23
2.2. Техніки адаптації зображень, шрифтів, блоків і навігації.....	24
2.3. Роль метатегів і типографіки для мобільних сайтів	26
2.4. Аналіз стандартів Google: Mobile-Friendly Test, Lighthouse, PageSpeed	28
РОЗДІЛ 3. ПРОЄКТНО-ПРИКЛАДНА ЧАСТИНА: РОЗРОБКА АДАПТИВНОГО ВЕБСАЙТУ	32
3.1. Постановка задачі та вимоги до сайту	32
3.2. Проектування структури та дизайну мобільного інтерфейсу	32
3.3. Вибір середовища розробки та технологій	33
3.4. Реалізація адаптивного макету: HTML, CSS, JS	35
3.5. Тестування сайту на різних екранах і пристроях	38
3.6. Перевірка швидкодії та юзабіліті: результати Lighthouse	41
3.7. Пропозиції щодо вдосконалення розробленого рішення	43
ВИСНОВКИ	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	49
ДОДАТКИ.....	56

ВСТУП

Зростання мобільного інтернет-трафіку продовжує перевищувати показники десктопного трафіку, що ставить перед розробниками вебсайтів завдання забезпечити гарний досвід взаємодії користувача з сайтом на смартфонах і планшетах. У своїй практиці сучасні компанії не можуть ігнорувати вимоги користувачів до швидкості завантаження, читабельності контенту та простоти управління жестами й тач-елементами. Навіть незначні затримки або незручності у навігації призводять до зростання показника відмов та зниження ключових бізнес-метрик: конверсії, часу сесії та лояльності клієнтів.

Особливості мобільних пристроїв – обмежений розмір екрану, різна щільність пікселів, відмінності в потужності браузерних рушіїв—вимагають відмінного підходу до верстки та програмної логіки порівняно з традиційним десктоп-дизайном. Суттєве значення для оптимального відображення має коректне налаштування мета-тегів і вибір «mobile-first» стратегії: базові стилі спочатку створюються під найменші екрани, а згодом через CSS-media-запити додаються поліпшення для ширших розмірів.

У цій роботі об'єктом дослідження було обрано вебресурс TurboVive - сайт сервісного центру з відновлення турбін. Цей проєкт має одночасно виконувати інформаційну, презентаційну й інтерактивну функції. Предметом дослідження є сукупність технологій і методик, за допомогою яких реалізується адаптивність інтерфейсу: семантична HTML-розмітка, структура CSS-стилів із розділенням на базові правила й брейк-поінти, а також мінімалістичні JavaScript-модулі для динамічної поведінки елементів.

Практична частина роботи починається з проєктування структури макета з використанням гнучких CSS Grid-сіток та лінійних Flexbox-контейнерів, оптимізованих під чотири типові розміри екранів. Для кожного з них створено відповідні медіа-запити: до 480 px, від 481 до 768 px, від 769 до 1199 px та понад 1200 px. Наступним кроком стало впровадження інтерфейсних скриптів, зокрема реалізація бургер-меню для мобільної навігації і плавних анімацій появи при

прокрутці. Останнім етапом було тестування працездатності всіх компонентів у середовищі Chrome DevTools з емуляцією пристроїв iOS та Android, а також проведення аудиту продуктивності, доступності й SEO через Google Lighthouse. Такий підхід дозволив перевірити коректність роботи адаптивних стилів і скриптів за реальних умов користування на різних пристроях. Наукова новизна роботи полягає у комплексному поєднанні семантичного підходу до HTML-розмітки з розширеною системою брейк-поінтів та мінімалістичними JavaScript-рішеннями, що дозволяють зменшити обсяг файлів і прискорити відображення контенту без втрати функціональності. Отримані результати демонструють, що навіть на слабких пристроях із повільним з'єднанням інтерфейс залишається чутливим та інтуїтивним.

Структурно робота складається з трьох розділів. Перший містить огляд сучасних принципів мобільного UX/UI та методів адаптації, другий - аналіз технічних засобів (CSS3-модулів, медіа-запитів, підходу mobile-first), третій - практичну реалізацію та результати тестування. У висновках наведено такі рекомендації щодо подальшої оптимізації інтерфйсу: впровадження прогресивного завантаження ресурсів, розширення набору брейк-поінтів та застосування PWA-технологій для офлайн-режиму.

РОЗДІЛ 1. ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ ОСНОВИ АДАПТАЦІЇ САЙТІВ ПІД МОБІЛЬНІ ПРИСТРОЇ

1.1. Актуальність адаптації сайтів у сучасних умовах

Стрімке зростання аудиторії смартфонів зробило адаптацію веб-сайтів під мобільні пристрої необхідною. Статистика підтверджує: близько 62% усього веб-трафіку нині припадає на мобільні телефони, що перетворює смартфони на основну платформу доступу до інформації [10]. Відповідно, ресурси мають бути пристосовані для перегляду на невеликих екранах, щоб охопити сучасну аудиторію.

Користувачі мобільного інтернету не терплять повільних або незручних сайтів. Якщо сторінка завантажується довше трьох секунд, близько 53% користувачів її покине [7]. Окрім швидкодії, важлива й якість інтерфейсу: користувачі очікують від мобільної версії такої ж зручності, як від десктопної; якщо ці очікування не виправдовуються, довіра до ресурсу падає – 57% опитаних не рекомендуватимуть компанію з поганим мобільним сайтом [43]. Таким чином, мобільна адаптивність прямо впливає на задоволеність, лояльність і репутацію.

Важливу роль відіграють і фактори SEO. Пошукові системи нині застосовують принцип mobile-first indexing – орієнтуються на мобільну версію сайту при індексації та ранжуванні. Враховуючи, що близько 65% глобальних пошукових запитів надходить зі смартфонів, відсутність мобільної версії може істотно знизити видимість ресурсу у пошуку. Швидкість і зручність мобільного сайту впливають на ранжування, повільний чи незручний для смартфона ресурс отримує нижчі позиції у видачі.

Мобільна адаптація тісно пов'язана з досвідом користувача (UX) і дизайном інтерфейсу (UI). Її основою є адаптивний вебдизайн: макет сторінки автоматично підлаштовується під розмір екрана, забезпечуючи читабельність тексту і зручну навігацію. Такий підхід вже став стандартом, що дозволяє одному сайту працювати на різних екранах. При цьому враховують специфіку мобільного UX:

достатній розмір кнопок для натискання пальцем, вертикальне розташування контенту, мінімізацію зайвих елементів. Зрозумілий і легкий у користуванні мобільний інтерфейс спонукає відвідувачів довше залишатися на сайті та знижує ймовірність швидкого відтоку [13].

З точки зору бізнесу, мобільна оптимізація впливає на конверсії та продажі. Якщо сайт незручний на смартфоні, користувачі не виконують цільових дій (наприклад, не здійснюють покупки чи замовлення) і компанія втрачає дохід. Мобільні пристрої вже забезпечують близько 60% онлайн-продажів, хоча конверсія на телефонах у середньому майже вдвічі нижча, ніж на ПК [35]. Водночас поліпшення мобільного досвіду дає відчутний ефект - скорочення часу завантаження сторінки на 1 секунду підвищило конверсію Walmart на 2% [7].

Попри переваги, адаптація сайтів під мобільні пристрої супроводжується викликами. Різноманітність моделей смартфонів і роздільних здатностей екранів змушує розробників тестувати ресурси для коректної роботи на будь-якому пристрої. Критичним є питання продуктивності: мобільні відвідувачі нерідко мають повільніше з'єднання, тож сайти повинні бути максимально оптимізовані для швидкого завантаження. Надалі мобільний сегмент лише зростатиме; вже з'являються нові типи пристроїв (наприклад, складані телефони), що потребують додаткової гнучкості дизайну, а очікування користувачів щодо швидкості та зручності зростають. Отже, мобільна адаптація залишається необхідною умовою успіху вебресурсу.

1.2. Еволюція користування інтернетом: перехід на мобільні пристрої

Як вже зазначалося попередньо, інтернет-технології за останні десятиліття стрімко еволюціонували, докорінно змінивши характер їх використання. Якщо в 1990-х роках доступ до мережі здійснювався переважно зі стаціонарних персональних комп'ютерів, то сьогодні основну частину інтернет-трафіку генерують мобільні пристрої. Тому, розглянемо ключові етапи розвитку користування Інтернетом – від ери стаціонарних ПК до домінування смартфонів і планшетів – з акцентом на зміни поведінки аудиторії, зростання мобільного

трафіку, розвиток бездротових технологій (3G, 4G, 5G) та вплив мобільних пристроїв на споживання цифрового контенту.

Отже, у 1990-х роках доступ до Інтернету відбувався переважно через стаціонарні ПК, підключені телефонною лінією (dial-up) зі швидкістю всього десятки кілобіт за секунду. На початку 2000-х комутовані з'єднання поступово витіснив широкосмуговий доступ (DSL, кабельний), що забезпечив значно вищі швидкості та постійне з'єднання. Це розширило можливості веб-контенту, проте користувачі все ще виходили онлайн тільки з фіксованого місця – дому або офісу.

Новий етап еволюції почався із впровадженням мобільного інтернету. Ранні мережі 2G (GPRS/EDGE) мали дуже низьку швидкість, аж поки на початку 2000-х не з'явилися мережі 3G, що вперше забезпечили мобільним телефонам доступ до Інтернету на швидкостях, до 1 Мбіт/с. Цей прорив підготував ґрунт для появи смартфонів. Презентація iPhone у 2007 році започаткувала сучасну еру смартфонів. У 2010-х смартфони на базі iOS та Android набули масового поширення, і мобільна аудиторія Інтернету зросла – станом на 2025 рік кількість користувачів смартфонів сягає 4,69 млрд осіб [4].

Черговий стрибок у можливостях бездротового доступу стався з упровадженням технології 4G (LTE) у 2010-х роках. Мережі четвертого покоління, вперше запущені наприкінці 2009 року, забезпечили мобільний широкосмуговий інтернет із набагато вищою швидкістю, ніж 3G. Це зробило можливим комфортне завантаження веб-сторінок і потокове відео зі смартфонів, що призвело до вибухового зростання мобільного трафіку. У 2016 році частка світового інтернет-трафіку з мобільних пристроїв досягла 51,3%, вперше перевищивши показник ПК (48,7%), тим самим закріпивши лідерство мобільного сегменту. Наприкінці 2010-х почалося розгортання мереж 5G (п'ятого покоління). Стандарт 5G забезпечує ще вищі швидкості та мінімальні затримки, відкриваючи нові можливості – наприклад, для масового підключення IoT. З розвитком 5G межа між «стаціонарним» і «мобільним» інтернетом зникає – високошвидкісний зв'язок доступний повсюдно [38].

Таким чином, мобільний інтернет зробив доступ до мережі постійним, легкодоступним і розповсюдженим: сучасні користувачі можуть перебувати онлайн майже безперервно, перевіряючи смартфон упродовж дня незалежно від місця. Нині понад половину загального часу в Інтернеті люди проводять саме зі смартфонів.

Можливості такого швидкого мобільного підключення спричинили бум потокового відео, музичних сервісів, соціальних мереж і мобільних додатків – наприклад, у другому кварталі 2021 року ~63% часу перегляду YouTube припадало на мобільні пристрої [8]. Більшість онлайн-сервісів тепер орієнтуються на мобільну аудиторію, адаптуючи свої продукти під екрани смартфонів. Планшети з більшими екранами полегшують читання та перегляд.

Смартфони та інші мобільні пристрої стали основним засобом доступу онлайн, перетворивши Інтернет на постійного супутника. Очікується, що подальший розвиток технологій (зокрема 5G) ще більше закріпить тенденцію до мобільності, роблячи доступ до цифрового світу менш залежним від місця та пристрою.

1.3. Типи мобільних пристроїв і роздільні здатності екранів

Сучасні мобільні пристрої різноманітні за своїм призначенням і розмірами екранів, але їх об'єднує ключовий компонент – дисплей. Від характеристик екрана (тип панелі, роздільна здатність, щільність пікселів, форм-фактор) залежить якість відображення інформації та зручність роботи користувача. У цьому підрозділі розглянемо класифікацію основних типів мобільних пристроїв (смартфони, планшети, смарт-годинники, електронні рідери тощо) та проаналізуємо технології дисплеїв, що в них використовуються (LCD, OLED, AMOLED, Retina, E-Ink).

Окрему увагу приділимо роздільним здатностям і щільності пікселів (PPI), форм-фактору екранів та їхньому впливу на адаптивний дизайн і розробку інтерфейсів. Наведемо актуальні технічні параметри й статистичні дані для підтвердження тенденцій розвитку дисплеїв.

Отже, роздільна здатність дисплея визначається кількістю пікселів по горизонталі та вертикалі екрану (ширина $\times$ висота у пікселях). Наприклад, роздільність 1920×1080 означає 1920 пікселів по ширині і 1080 по висоті – такий формат також відомий як Full HD 1080p. У цифрових пристроях, які мають фіксовану матрицю пікселів, часто під «роздільною здатністю» розуміють саме ці габаритні розміри матриці. Водночас технічно правильніше говорити про роздільну здатність як про густину (щільність) пікселів, тобто кількість пікселів на одиницю довжини (або площі). Показником цього є ppi (pixels per inch, пікселів на дюйм) – число пікселів, яке припадає на один дюйм довжини екрану. Щільність ppi пов'язана як з фізичним розміром дисплея, так і з його роздільністю: два екрани з однаковою кількістю пікселів матимуть різну щільність, якщо їх діагоналі різні (на більшому екрані пікселі розподіляться «рідше», і зображення виглядатиме менш різким).

У мобільних пристроях досягнуті дуже високі роздільні здатності на відносно малих діагоналях, що забезпечує велику щільність пікселів. Для оцінки: 5-дюймовий смартфон з HD-екраном (1280×720) має ~ 294 ppi, такий самий з Full HD (1920×1080) – вже ~ 440 ppi, а з 4K UHD (3840×2160) – понад 800 ppi. Як згадувалося, практичний поріг, після якого зростання ppi мало впливає на чіткість для ока, становить близько 300 ppi для телефону на звичній відстані використання. З розвитком технологій виробники перевищили цей поріг: наприклад, ранні “Retina” смартфони мали ~ 326 ppi, а нові компактні флагмани з 1080p дисплеями – біля 450–500 ppi. Теоретично око здатне розрізнити покращення, якщо піднести екран ближче або мати надзвичайно гострий зір, але в нормальних умовах різниця між 400 і 500 ppi майже невідчутна. Для планшетів і ноутбуків, що розташовуються далі від очей, достатньою є навіть менша щільність (200–300 ppi), аби досягти «ретино-ефекту». Таким чином, інженери орієнтуються на компроміс: з одного боку, висока роздільна здатність забезпечує плавність і деталізацію зображення, з іншого – збільшує вимоги до графічного процесора та енергоспоживання.

Адже чим більше пікселів потрібно опрацювати, то більшим є навантаження на GPU і швидше розряджається батарея. Збільшення щільності пікселів понад оптимальний рівень може не давати помітного виграшу в якості, проте зменшує автономність пристрою. Практично всі виробники це враховують: наприклад, смартфони з 4К-дисплеями часто за замовчуванням рендерять інтерфейс у більш низькій роздільності, вмикаючи 4К лише для відео чи фото, щоб ощадити ресурс.

Форм-фактор екрана охоплює геометричні параметри дисплея – діагональ, співвідношення сторін, кривизну тощо. Різні категорії пристроїв мають свої типові форм-фактори. Смартфони еволюціонували від компактних 3,5–4" дисплеїв 3:2 або 16:9 до сучасних безрамкових екранів 6–7" із видовженим співвідношенням (18:9, 19.5:9 і більше). Середній розмір екрану смартфона нині становить приблизно 6,3 дюйма, при цьому виробники прагнуть збільшити корисну площу без надмірного збільшення габаритів самого пристрою (завдяки тоншим рамкам і вирізам під камеру). Планшети історично поділяються на компактні (~7–8") та повнорозмірні (~10–13"); співвідношення сторін у них варіює від 4:3 (краще для читання, вебу) до 16:10 чи 16:9 (краще для відео). Смарт-годинники, як зазначалося, можуть бути квадратними (наприклад, 368×448 пікселів у Apple Watch) або круглими (наприклад, 454×454 у деяких моделей Garmin) – це диктує різну організацію інтерфейсу. Аспекти співвідношення сторін впливають на представлення інформації: на широких екранах зручно розміщувати елементи поруч, тоді як на вузьких і витягнутих – доводиться більше прокручувати контент по вертикалі. З'явилися гнучкі та складані дисплеї, які можуть змінювати форму (смартфон розкладається в міні-планшет) – для них інтерфейси повинні автоматично перебудовуватися під різні розміри вікна.

Тож бачимо, що такий різнобарвний ландшафт розмірів і форматів екрану висуває суворі вимоги до адаптивного дизайну інтерфейсів. Веб-розробка вже давно перейшла до концепції *responsive design* - макети сайтів реагують на ширину екрану і перебудовуються під неї. Немає можливості оптимізувати веб-сторінку лише під одну роздільність, адже в глобальній статистиці жоден розмір

екрана не домінує: наприклад, найпоширеніший розмір серед мобільних пристроїв (близько 360×800 точок) займає лише $\sim 10\%$ ринку, а наступні за ним 390×844 , 393×873 , 375×812 тощо мають частки 4-6% кожен. Отже, дизайнери закладають кілька ключових «брейкпойнтів» (типових точок перелому макета) - для маленьких смартфонів, для великих телефонів/малих планшетів, для планшетів у ландшафтному режимі тощо. Наприклад, на смартфоні меню може трансформуватися в випадаючий список, тоді як на широкому планшеті ті самі пункти показані у вигляді панелі навігації. Для нативних мобільних застосунків застосовуються аналогічні принципи адаптивності: розробники проєктують інтерфейс з урахуванням різних розмірів екрану і щільності.

Смартфони.

Смартфон - портативний телефон з функціональністю кишенькового комп'ютера. Смартфони обладнані сенсорними екранами діагонально здебільшого 5-7 дюймів, що дозволяє керувати інтерфейсом дотиком. Для сучасних смартфонів характерні високі роздільні здатності дисплея (щонайменше HD або Full HD), а флагманські моделі оснащуються навіть QHD ($1440p$) чи 4K екранами. Наприклад, смартфон Sony Xperia 1 має 6,5-дюймовий 4K OLED-дисплей з роздільністю 3840×1644 , що дає надзвичайно високу щільність ~ 643 ppi [12]. Типові значення щільності пікселів для телефонів сьогодні знаходяться в діапазоні 300–500 ppi, що забезпечує дуже чітке зображення. Більшість сучасних смартфонів використовують OLED-екрани (або їх різновид AMOLED) завдяки кращій контрастності та енергоефективності. За даними досліджень, у 2024 році частка OLED досягла понад 50% ринку екранів смартфонів, вперше випередивши LCD [15]. Це означає, що купуючи новий смартфон, користувач з більшою ймовірністю отримає модель з OLED-дисплеєм, ніж з LCD, оскільки технологія OLED стрімко перейшла з флагманського сегмента в середній клас. Смартфони мають похилу орієнтацію використання (портретний режим) і широкий діапазон співвідношень сторін екрана – від класичного 16:9 до витягнутих 19.5:9 чи навіть 21:9 у деяких моделей. Це

створює виклики для дизайнерів інтерфейсів, адже потрібно забезпечити коректне відображення на різних за формою та розміром дисплеях.

Планшети.

Планшети - портативні комп'ютери з більшим екраном (зазвичай 7-13 дюймів), які займають проміжне місце між смартфонами та ноутбуками. Вони зручні для веб-серфінгу, читання, мультимедіа та роботи з документами. Екрани планшетів часто виконані за технологією LCD (IPS-панелі для широких кутів огляду), хоча деякі преміум-моделі застосовують AMOLED-дисплеї. Роздільна здатність у планшетів, як правило, не поступається смартфонам за абсолютною кількістю пікселів (наприклад, 1920×1200 чи 2560×1600), але через більшу діагональ щільність пікселів може бути нижчою – типово 150-300 ppi. Так, популярний 10-дюймовий планшет із дисплеєм 1920×1200 матиме ~224 ppi, тоді як 9,7-дюймовий iPad із Retina-дисплеєм 2048×1536 має щільність 264 ppi (Apple вважає це достатнім для ока на відстані використання планшета). Планшети часто використовуються в альбомній орієнтації та можуть мати співвідношення сторін 4:3 (характерно для iPad, зручніше для читання) або 16:10 (поширено в Android-планшетах, зручніше для відео). Більший фізичний розмір екрана дозволяє відображати більше контенту, однак вимагає адаптації інтерфейсу – наприклад, на планшеті може бути реалізовано багатоколонкове розташування елементів, недоцільне на смартфоні.

Смарт-годинники.

Смарт-годинники - носимі пристрої у формі годинника, оснащені малим сенсорним екраном (зазвичай діагоналлю 1–2 дюйми). Вони працюють під управлінням спеціальних операційних систем для носимих гаджетів і часто синхронізуються зі смартфоном. Екрани смарт-годинників можуть бути круглими або прямокутними; приклад – Apple Watch із прямокутним дисплеєм ~1,78" (компактний форм-фактор). Через малі розміри щільність пікселів дуже висока – близько 300 ppi і більше, що дозволяє відображати текст і графіку з належною різкістю навіть на крихтному дисплеї. Переважна більшість сучасних розумних годинників оснащена OLED/AMOLED-дисплеями, оскільки вони

забезпечують яскраві кольори, високий контраст і можливість постійно ввімкненого циферблату (режим Always-On) без значного впливу на автономність. OLED-матриця дозволяє вибірково підсвічувати окремі пікселі (наприклад, тільки цифри годинника), залишаючи фон чорним і енергоощадним. Деякі спеціалізовані спортивні моделі годинників застосовують технологію transfective LCD або Memory-in-Pixel LCD, що теж дає змогу економити заряд батареї та добре читатися на сонці, але такі екрани зазвичай мають тьмяніші кольори. Смарт-годинники ставлять особливі вимоги до дизайну інтерфейсів. Інформація має бути лаконічною, шрифти достатньо великі, а елементи інтерфейсу зручні для торкання на дуже обмеженій площі екрана.

Електронні рідери.

Електронні рідери (букрідери) - це пристрої для читання електронних книг, основною особливістю яких є використання дисплея на основі електронного паперу (E-Ink). Діагональ екрану е-рідерів зазвичай становить 6-8 дюймів, рідше - до 10" у більших моделях. Екран E-Ink кардинально відрізняється від LCD/OLED: він не випромінює світло, а відбиває зовнішнє (як звичайний папір). Завдяки цьому читати з такого дисплея комфортно навіть при яскравому сонячному освітленні, а навантаження на очі мінімальне. Електронне чорнило є бістабільним - зображення зберігається без споживання енергії і оновлюється лише при перегортанні сторінки. Це забезпечує тижні роботи від батареї без підзарядки. Роздільна здатність сучасних E-Ink екранів достатньо висока: типово ~1024x758 або 1448x1072 для 6" дисплеїв, що відповідає 300 ppi. Така щільність пікселів дає якість тексту, порівнянну з друком на папері (для порівняння, 167 ppi в старіших моделях Kindle вже вважається недостатнім рівнем). Більші електронні рідери (напр. 10-дюймові) можуть мати трохи меншу щільність (~200–250 ppi) при все ще прийнятній чіткості. Більшість е-рідерів відображають 16 відтінків сірого; існують моделі з кольоровим E-Ink (технології типу Kaleido або Gallery), але вони поки що поступаються традиційним дисплеям у насиченості кольорів і швидкості оновлення. Електронні рідери призначені переважно для статичного контенту, тому низька частота оновлення (зазвичай ~1

кадр на секунду при перегортанні) не є проблемою, зате відсутність мерехтіння і підсвічування робить читання дуже комфортним.

Тож, бачимо, що завдяки впровадженню responsive-підходів і високим стандартам «рети́на-яко́сті» сучасні користувачі можуть насолоджуватися комфортним та естетичним відображенням інформації на екранах своїх мобільних пристроїв за будь-яких умов, не зважаючи на форму і розміри.

1.4. Поняття адаптивного та респонсивного дизайну

Таким чином, маємо розуміння того, що сучасні сайти повинні коректно працювати на будь-яких пристроях. Існують два основні підходи до цього: респонсивний (відгукливий) дизайн і адаптивний дизайн.

Респонсивний вебдизайн - підхід, за якого веб-сторінка має єдиний гнучкий макет, що динамічно підлаштовується під різні розміри екрану користувача. Респонсивний сайт використовує гнучкі сітки та CSS-медіазапити, щоби автоматично масштабувати текст, зображення й інші елементи під поточну ширину екрану, зберігаючи зручність користування на будь-якому пристрої [14].

Адаптивний вебдизайн передбачає розробку кількох фіксованих макетів сторінки, кожен з яких оптимізований під певний діапазон розмірів або тип пристрою. На практиці зазвичай створюються окремі дизайни для типових форм-факторів (смартфон, планшет, настільний екран тощо), між якими сайт перемикається залежно від пристрою користувача [14].

Обидва підходи мають на меті забезпечити зручність перегляду на різних пристроях, проте їхні механізми відрізняються. Респонсивний дизайн базується на одному гнучкому макеті, що реагує на зміну ширини екрану, тоді як адаптивний використовує набір заздалегідь розроблених шаблонів для різних пристроїв. Відповідно, кожен метод має свої переваги й недоліки. Responsive підхід забезпечує однаковий користувацький досвід на всіх платформах і потребує менше зусиль на підтримку (оновлюється один сайт). Також він сприятливіший для SEO, адже пошукові системи краще індексують єдину мобільно-орієнтовану версію ресурсу. Серед недоліків респонсивного дизайну -

необхідність дизайнерських компромісів для різних екранів і більші початкові витрати часу на розробку універсального макету.

Адаптивний підхід дає змогу краще оптимізувати інтерфейс та контент під кожен категорію пристроїв, що часто забезпечує кращу швидкодію і зручність користування на цільових платформах. Натомість адаптивний дизайн є відчутно складнішим і дорожчим у реалізації, адже потрібно створювати й підтримувати кілька версій сайту одночасно, що утруднює масштабування під нові пристрої і вимагає більше ресурсів на тестування та супровід проекту.

Респонсивний дизайн за останнє десятиліття став стандартом веброзробки, оскільки його універсальність у підтримці різних пристроїв зумовила широке впровадження цього підходу. Крім того, респонсивний сайт потребує менше зусиль при появі нових нестандартних екранів – макет автоматично адаптується без додаткової розробки.

Проте, саме адаптивний підхід залишається актуальним у певних ситуаціях. Наприклад, для складних інтерактивних веб-застосунків або великих сайтів, спочатку не орієнтованих на мобільні пристрої, адаптивний дизайн забезпечує більш точну оптимізацію інтерфейсу під можливості конкретних платформ. Отже, хоча в сучасних проектах переважає респонсивність, розуміння обох підходів дозволяє обрати оптимальний метод розробки залежно від потреб проекту.

1.5. Основні принципи UX/UI для мобільних інтерфейсів

Користувацький досвід (UX) та дизайн інтерфейсу користувача (UI) є ключовими чинниками успіху мобільних сайтів та застосунків. UX відповідає за загальне враження користувача – зручність, задоволення, простоту взаємодії, а UI – за візуальне представлення та взаємодію з елементами інтерфейсу. Як вже зазначалося вище, недостатньо продуманий UX на мобільних пристроях може призвести до втрати більшості користувачів: 88 % користувачів не повернуться на сайт після поганого користувацького досвіду, тоді як близько 74 % користувачів радше повернуться на сайт, адаптований під мобільні пристрої [23].

Таким чином, інтуїтивний UX/UI напряду впливає на ефективність ресурсу – покращує конверсію, підвищує лояльність та задоволеність користувачів.

Ясна візуальна ієрархія допомагає користувачам швидко зорієнтуватися в інтерфейсі. За словами спеціаліста Nielsen Norman Group, *«чітка візуальна ієрархія спрямовує погляд до найважливіших елементів на сторінці»* [34]. На практиці це означає використання різних розмірів, кольорів, контрасту та групування для позначення пріоритетності контенту. Заголовки та найважливіші кнопки можна зробити більшими або яскравішими, відокремлювати їх просторами (white space), щоб вони «виділялися», а менш важливий контент розміщувати нижче або в інших розділах. Наприклад, усі кнопки дій (Call to Action) часто роблять контрастного кольору, щоб вони кидалися в очі, а довгі тексти ділять на декілька блоків з підзаголовками. При цьому дизайн повинен бути послідовним у всьому мобільному інтерфейсі: подібні дії повинні мати схожий вигляд, що сприяє інтуїтивному розумінню (наприклад, усі посилання виділені підкресленням чи кольором). Завдяки чіткій ієрархії користувач одразу розуміє, куди треба дивитися і які елементи інтерфейсу є ключовими для виконання завдань.

Мобільний інтерфейс має враховувати особливості роботи пальцем. Щоб користувачі могли легко натискати елементи, інтерактивні цілі повинні бути достатньо великими та розташовані з урахуванням великого пальця. Стандарти доступності (WCAG 2.1) рекомендують встановлювати область дотику не менше 44×44 CSS-пікселів для надійного натискання [48]. Менші цілі важко влучити пальцем, особливо людям з обмеженою моторикою. Також важливим є правильне використання масштабування. Уникають фіксованих ширин та великих елементів, що можуть виходити за межі екрана.

Особлива увага приділяється контрастності і читаності тексту: WCAG рекомендує співвідношення контрастності фону і тексту принаймні 4.5:1 для звичайного тексту (рівень AA), щоб користувачам з порушенням зору було зручно читати з екрану. Для великих заголовків допускається трохи нижча контрастність (3:1). Крім того, рекомендується не використовувати дрібні або

схожі між собою символи: наприклад, цифра «0» має відрізнятися від великої латинської «O», а букви «l» і «I» – бути різними. Оптимальний розмір шрифту тіла тексту на мобільному інтерфейсі – не менше 16 пікселів, а заголовки приблизно у півтора рази більші. Текст не повинен бути надто маленьким, щоб його можна було зачепити пальцем, але й не настільки великим, щоб займати весь екран. Важливо обирати чіткі контрастні кольори та читабельні шрифти (зокрема, без зарубок) для головного тексту. Усе це підвищує доступність та комфорт користувача.

Колірна палітра мобільного застосунку чи сайту також впливає на сприйняття та ефективність. Для гарної ієрархії часто використовують яскраві або насичені кольори для наголосу та нейтральні тони для фону або другорядних елементів. Не слід використовувати багато різних шрифтів чи кольорів одночасно – це створює візуальний шум. Як правило, застосовують два шрифти: один для заголовків, інший для основного тексту, з декількома стилями (звичайний, жирний тощо) для розмежування рівнів інформації. Надто легкі обводки шрифтів слід уникати на фоні екрану через знижену читабельність. Тож, бачимо, що загалом, дизайн шрифту має бути оптимізований під швидке сприйняття інформації «на ходу».

1.6. Огляд сучасних технологій, методів та фреймворків адаптації

Media-запити і гнучкі сітки

Основним інструментом у адаптивній верстці є *CSS Media Queries* – правила, що застосовують стилі за умови виконання певного медіакритерію (найчастіше ширина або орієнтація екрану). MDN відзначає: «*медіазапити* – ключова частина респонсивного дизайну, оскільки вони дозволяють створювати різні макети залежно від розміру вікна перегляду» [27]. Тобто, наприклад, якщо ширина екрану менша за 600px, до блоку застосовуються одні стилі, а якщо більше – інші. Разом із медіазапитами часто використовують гнучкі (флюїдні) сітки – *relative units* (відсотки, *em/rem*) замість фіксованої ширини в пікселях.

Наприклад, ширина колонки робиться width: 50%, а не width: 600px. Це дозволяє віджетам «підтягуватися» до доступного простору.

Еталонні практики responsive design згруповані в три основні принципи: *флюїдні сітки (fluid grids)*, *гнучкі зображення (flexible images/media)* та *медіазапити*. Наприклад, зображення можна задавати у відсотках або встановлювати max-width:100%, щоб воно не виходило за межі блоку. Також медіазапити дозволяють змінювати макет з одноколонного на багатоколонний на більших екранах, змінювати відступи та розміри шрифтів залежно від розміру вікна.

Підхід Mobile-First та прогресивне покращення

Стратегія *Mobile-First* полягає в тому, щоб спочатку розробляти макет для малих екранів, а потім поступово «нарощувати» його для більших. UXPin радить починати з найменшого роздільного дозволу і далі «підтягувати» дизайнерські рішення, щоб уникнути обмежень контентом [45]. Відомий дизайнер Люк Вроблевські описував mobile-first як філософію, яка дозволяє охопити більше людей і змушує фокусуватися на основному контенті [41]. Тобто розробник починає з базової мобільної версії (де лишаються найважливіші функції) і лише потім додає додаткові можливості, стилі чи елементи для планшетів і десктопів.

Такий підхід органічно поєднується з *прогресивним покращенням* (Progressive Enhancement): спочатку надається функціональність і контент для всіх користувачів на базовому рівні, а сучасні браузері отримують «покращення» у вигляді додаткових стилів та скриптів. MDN визначає прогресивне покращення як методологію, за якої «першочергово забезпечується базовий контент і функціональність для максимально широкої аудиторії, а згодом – найкращий можливий досвід для користувачів найсучасніших браузерів» [29]. Таким чином, mobile-first — це застосування прогресивного покращення від найпростішого (малий екран, найповільніше з'єднання) до більш складного дизайну для потужніших пристроїв.

Не можна забувати і про мета-тег `<meta name="viewport" content="width=device-width, initial-scale=1.0">`. Він інструктує браузер брати за

базову ширину екрану пристрою, що дозволяє верстці «відгукуватися» на реальні розміри вікна. Без цього тегу мобільні браузері часто масштабують сторінку за замовчуванням, що призводить до «жахливих» мініатюрних текстів і появи горизонтальної прокрутки. Встановлення `width=device-width` гарантує, що 1 CSS-піксель відповідає 1 фізичному пікселю екрану, а `initial-scale=1.0` встановлює початковий масштаб 100 %, що є основою для побудови адаптивного макету. Таким чином, поєднання коректного мета-тегу, медіазапитів та гнучких сіток є фундаментом для сучасної адаптивності веб-інтерфейсів.

1.7. Порівняльний аналіз успішних мобільних адаптацій

Для порівняльного аналізу та оцінки поєднаємо теоретичний аналіз із реальними кейсами, виділивши такі ключові критерії:

- швидкість завантаження: якщо сторінки вантажаться понад 3 с, близько 53% користувачів її покидають; зменшення часу на 0,1 с може збільшити конверсію на 8,4%;
- ефективність UX/UI: простий інтерфейс і чітка ієрархія (помітний заголовок, контрастні кольори) спрямовують увагу користувача;
- структура контенту: сторінки містить лише необхідну інформацію, розміщену послідовно на одній сторінці з єдиним СТА.

Нижче наведено порівняльний аналіз того, як два великі бренди — McDonald's Україна та Adidas Україна — адаптували свої сайти під мобільні пристрої за ключовими критеріями: масштабу вікна (viewport), навігація, структура контенту, візуальна ієрархія, СТА-елементи та швидкість завантаження.

Масштабування та основа адаптивності

McDonald's і Adidas обидва використовують мітатеґ `<meta name="viewport" content="width=device-width, initial-scale=1">`, що гарантує відповідність макета фізичним розмірам екрану пристрою. Цей тег є базовим елементом будь-якого RWD-проекту, оскільки без нього медіазапити не вступають в дію, а сторінка масштабуватиметься застарілим способом [32].

Навігація

McDonald's застосовує «бургер-меню», яке висувається зверху та займає весь екран при відкритті. Усередині — вертикальний список категорій (Сніданок, Бургери тощо) та секції додаткових сервісів (McDelivery, мобільний застосунок). Такий підхід економить місце й дозволяє користувачу швидко зорієнтуватися в каталозі [26].

Adidas використовує фіксований хедер із піктограмою «бургера», проте при відкритті меню воно розгортається як вертикальна панель із вкладеними підкатегоріями (Чоловіки → Взуття → Кросівки тощо). У підкатегоріях реалізовано акордеон-стиль, що полегшує пошук конкретних лінійок товарів [2].

Структура контенту та вміст «понад фолдом»

McDonald's зверху демонструє карусель із великими рекламними слайдами (новинка МакКріспі, НайсПрайс), кнопками «Детальніше», а далі — пропозиції щодо завантаження мобільного застосунку. Такий послідовний потік блоків відповідає моделі AIDA: увага (яскраві слайди) → інтерес (опис акцій) → бажання (завантажити застосунок) → дія (кнопка переходу) [26].

Adidas у «першому екрані» мобільної версії відображає промо-банери з умовами безкоштовної доставки та акціями, нижче — блоки категорій («Чоловіки», «Жінки», «Діти») з картками лінійок (Originals, Superstar тощо). Цей дизайн орієнтований на швидкий доступ до каталогу товарів за допомогою іконок і зрозумілих підписів [2].

Візуальна ієрархія та UX/UI

McDonald's робить акцент на великих фонових зображеннях із текстом-оверлеєм, виділяє кнопки «Детальніше» яскравим кольором, а логотип і пошук фіксуються зверху, що підвищує впізнаваність бренду й скорочує час пошуку [26].

Adidas використовує монохромні іконки та достатній білий простір навколо, щоб не перевантажувати екран. Кожний ключовий блок (наприклад, «Бестселери») розташовано з вирівнюванням на сітці, що полегшує сканування зору та швидкий вибір категорії [2].

СТА-елементи

McDonald's пропонує кілька СТА: «Детальніше» під кожним слайдом, кнопки завантаження застосунку (App Store, Google Play) та посилання на «Знайти нас» (геолокація ресторанів) [26].

Adidas розставляє один пріоритетний СТА-елемент «Дізнатися більше» під банером акції та інтегрує кнопки «Перейти до категорії» напроти кожної лінійки. Ці кнопки великі, в контрастному кольорі, відповідають стандартам сенсорного введення [2].

Швидкість завантаження

McDonald's оптимізує зображення через WebP та lazy-loading, мінімізує сторонні скрипти (наприклад, аналітику активують після першого екрану).

Adidas застосовує CDN та віддає пріоритет критичним CSS, але водночас завантажує більше шрифтів та банерів, що іноді може трохи уповільнювати «перший контентний фарстраф» (First Contentful Paint).

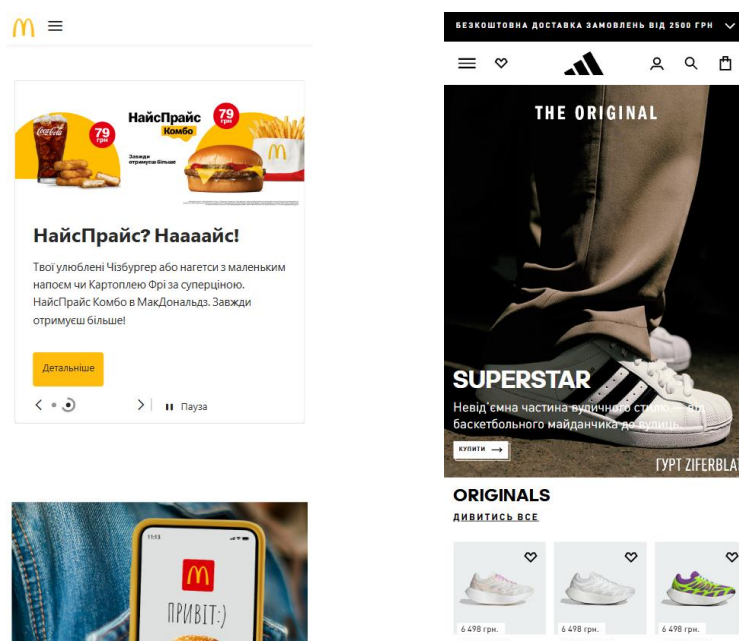


Рис.1.1. Мобільні версії сайтів McDonalds і Adidas

Отже, зведемо підсумки у таблицю основних критеріїв порівняльного аналізу адаптацій.

Таблиця 1.1. Порівняльний аналіз адаптацій

Критерій	McDonald's	Adidas
Метатег viewport	Присутній, стандартний	Присутній, стандартний
Меню навігації	Бургер-меню	Акордеон-меню
Контент “понад фолдом”	Промо-карусель + СТА на застосунок	Акційні банери + категорії товарів
Візуальна ієрархія	Фонові зображення + яскраві кнопки	Сітка + білий простір
СТА	Мультикнопки (детальніше, геолокація)	Один пріоритет + кнопки категорій
Завантаження	2 с (оптимізація WebP, lazy-load)	3–4 с (менше lazy-load, більше ресурсів)

Як висновок бачимо, що обидва сайти дотримуються сучасних стандартів адаптивного дизайну, проте McDonald's робить сильний акцент на швидкість та простоту просування мобільного застосунку, тоді як Adidas зосереджується на зручності покупки та широкому каталозі, жертвуючи трохи швидкістю завантаження. Незважаючи на це, обидва бренди успішно утримують високу конверсію завдяки продуманим UX/UI-рішенням і чіткій візуальній подачі.

РОЗДІЛ 2. СИСТЕМНО-АНАЛІТИЧНЕ ДОСЛІДЖЕННЯ ТЕХНІЧНИХ РІШЕНЬ АДАПТАЦІЇ

2.1. Роль CSS3 та media-запитів у формуванні адаптивності

CSS3 (Cascading Style Sheets Level 3) суттєво сприяв поширенню адаптивного дизайну. Зокрема, у CSS3 з'явилися media-запити – правила *@media*, що дозволяють умовно застосовувати CSS-стилі в залежності від характеристик пристрою. Media-запити стали базовим інструментом респонсивного веб-дизайну, дозволяючи коригувати макет та стилі сторінки під ширину екрану, орієнтацію, роздільність та інші параметри. У сукупності з іншими можливостями CSS3 (гнучкими макетами та відносними одиницями) це гарантує гнучкість інтерфейсів незалежно від розміру екрану.

Media-zanum – це правило CSS, яке дозволяє умовно застосовувати блок стилів тільки при виконанні певної умови. Він задається через *@media* з медіа-типом (наприклад, *screen* чи *print*) і одним або кількома виразами («*media features*»), що описують властивості пристрою. Наприклад, правило *@media (max-width: 768px)* застосує стилі лише на екранах шириною 768px або менше [30]. Умови можна поєднувати, задаючи одночасно ширину, орієнтацію, роздільність та інші характеристики. Таким чином, media-запити дозволяють вивести різні стилі для конкретних категорій пристроїв (смартфонів, планшетів, десктопів тощо), що й робить їх ключовим інструментом адаптивної верстки.

Найпоширеніші умови (*media features*) у медіа-запитах:

- *min-width*, *max-width* – мінімальна/максимальна ширина області перегляду (*viewport*). Зазвичай використовуються для «точок розриву» макету;
- *min-height*, *max-height* – мінімальна/максимальна висота області перегляду;
- *orientation* – орієнтація екрану (*portrait* чи *landscape*). Ця умова дозволяє адаптувати дизайн для книжкової чи альбомної орієнтації;
- *resolution* – роздільна здатність екрану (наприклад, *min-resolution* у *dpi* або *dppx*).

Також існують інші умови: aspect-ratio, device-width, color. Ці умови дають змогу налаштовувати адаптивність у специфічних випадках [30].

Сучасні вебсторінки, оформлені за допомогою CSS3 і media-запитів, автоматично підлаштовуються під будь-які розміри екранів, забезпечуючи оптимальне відображення контенту та зручний користувацький досвід. Поєднання гнучких макетів, відносних одиниць і медіа-запитів дає змогу отримати універсальний дизайн, який однаково добре працює на різних пристроях. Це виключає потребу створювати окремі версії сайту і забезпечує єдиний коректний вигляд інтерфейсу на всіх платформах. Таким чином, CSS3 і media-запити є фундаментом адаптивної верстки сучасних вебінтерфейсів, що гарантує гнучкість і зручність у поданні контенту користувачам незалежно від розміру їхніх екранів.

2.2. Техніки адаптації зображень, шрифтів, блоків і навігації

Мультимедійний контент, особливо зображення, часто складає основну частку (до 70%) трафіку веб-сторінки, тому його оптимізація критично важлива [28]. Адаптивний підхід дозволяє підбирати зображення відповідного розміру та формату залежно від характеристик пристрою (екран, роздільна здатність тощо), що суттєво підвищує продуктивність. Розглянемо детальніше оптимальні техніки для розв'язання головних завдань з адаптації.

Так, HTML надає атрибути srcset і sizes, завдяки яким браузер вибирає оптимальне зображення з кількох варіантів для різних ширин екрану.

Атрибути srcset та sizes – дозволяють вказати декілька версій одного зображення для різних розмірів екрана, щоб браузер обирав найкращий варіант залежно від параметрів дисплея [28].

Формат WebP – сучасний формат зображень із кращою компресією, ніж JPEG/PNG. Для його доставки використовують <picture> зі джерелом type="image/webp", а для браузерів без підтримки WebP – звичайне зображення.

Ліниве завантаження (lazy loading) – атрибут loading="lazy" дозволяє відкласти завантаження зображень, які знаходяться за межею видимості, до

моменту прокрутки. Це зменшує початкове навантаження сторінки та пришвидшує її рендеринг.

Для масштабування тексту використовують відносні одиниці виміру: `em` та `rem` (що базуються на розмірі шрифту поточного або кореневого елемента), а також `vw/vh`, прив'язані до розмірів вікна перегляду. Такий підхід забезпечує пропорційне масштабування шрифтів і блоків залежно від ширини пристрою. Наприклад, одиниця `1vw` завжди дорівнює 1% ширини вікна, тому текст плавно змінюється разом із розміром екрана.

Новим інструментом стала функція `CSS clamp()`, яка обмежує значення в діапазоні між заданим мінімумом та максимумом. Наприклад, `font-size: clamp(1rem, 2.5vw, 2rem)` означає, що розмір шрифту зростає із шириною екрану, але не стає меншим за `1rem` і не більшим за `2rem`. Це забезпечує плавне масштабування тексту без необхідності застосовувати багато медіа-запитів.

Важливо виділити механізми адаптивного компоновання блоків, оскільки сучасні CSS-макети для адаптивного дизайну опираються на `Flexbox` і `Grid`.

`Flexbox` — одновимірний механізм для розташування елементів у ряд чи стовпець із гнучким розподілом простору. `CSS Grid` — двовимірний механізм, що розбиває макет на рядки й колонки, полегшуючи створення складних адаптивних сіток. Для умовної зміни стилів залежно від ширини екрану застосовують медіа-запити — вони виконують перевірки (наприклад, ширини вікна) і застосовують різні CSS-правила. Цей підхід дозволяє, наприклад, задавати одноколонковий макет для вузьких екранів і багатоколонковий — для широких.

Важливим компонентом в адаптації навігації для мобільних пристроїв є «бургер-меню» — іконка з трьома паралельними смужками, натискання на яку розгортає повне меню. Використовується для компактного представлення навігації на мобільних пристроях [25].

Акордеон-меню — меню з розділами, які можна розгортати або згорнути. Така схема дозволяє приховати великі обсяги контенту (наприклад, підпункти меню) і виводити їх лише за потреби користувача.

Фіксовані панелі – навігаційні блоки, закріплені у верхній чи нижній частині екрана, що залишаються видимими при прокрутці. Це гарантує постійний доступ до ключових пунктів меню незалежно від положення на сторінці.

2.3. Роль метатегів і типографіки для мобільних сайтів

Метатеги та типографіка відіграють ключову роль у забезпеченні зручності перегляду контенту на екранах різних розмірів. Також розглядаються принципи сучасної веб-типографіки для мобільних: вибір шрифтів, розміри та міжрядковий інтервал, контрастність та адаптивне масштабування тексту. Грамотне використання цих технологій покращує читабельність, навігацію й ефективність мобільних сайтів.

Метатеги та елементи `<head>` відіграють роль у SEO та зручності користування. Тег `<title>` задає назву сторінки, яка відображається у вкладці браузера та результатах пошукової видачі. Зміст заголовка є одним із ключових сигналів для алгоритмів пошукових систем та першим «гачком» для користувачів у результатах пошуку. Рекомендується формувати `<title>` з урахуванням релевантних ключових слів та зберігати довжину приблизно до 60 символів, щоб уникнути обрізання в SERP. Метаопис (`<meta name="description">`) дає стислий опис вмісту сторінки. Хоча безпосередньо він не впливає на позиції в пошуку, хороший опис підвищує клікабельність (CTR) результату.

Мета-тег `charset` (наприклад, `<meta charset="UTF-8">`) визначає кодування символів документа. Використання універсального кодування UTF-8 гарантує коректне відображення тексту будь-якими мовами (у тому числі українською), запобігаючи «кракозябрам» та проблемам із відображенням національних символів. Загалом, продумана конфігурація цих метатегів полегшує користувацький досвід (зручний перегляд і навігацію) та покращує індексацію сайту пошуковими роботами.

У виборі шрифтів для забезпечення гарної читабельності мобільних сайтів зазвичай обирають прості, чіткі накреслення. Сучасні практики рекомендують використовувати санс-серіфні шрифти (без зарубок) або легко читабельні з

високим «x-height». Такі шрифти мають чистий дизайн і добре сприймаються на невеликих екранах. Прикладом є шрифт Roboto (геометричний санс-серіф від Google), який розроблений для чіткого відображення на Android-пристроях [22]. Якщо ж використовуються шрифти з зарубками (serif), це повинні бути добре структуровані варіанти з відкритими формами букв, як-от Georgia. Слід уникати надмірно декоративних накреслень (староанглійський стиль, тіні, орнаменти тощо), оскільки вони дуже ускладнюють читання на малих розмірах.

Щоб забезпечити комфортне читання, основний текст мобільного сайту слід робити достатньо великим. Загальновідоме правило – це ~16 px для body text. Менші розміри можуть призводити до напруження очей і погіршення доступності для людей з ослабленим зором. Заголовки у мобільній версії зазвичай є більшими за основний текст, але слід слідкувати, щоб лінії не виходили за межі екрану. За дослідженнями, оптимальна довжина рядка – близько 50–75 символів, тому розмір шрифту слід підбирати з урахуванням ширини вікна переглядача.

Прийнятний міжрядковий інтервал покращує читабельність, особливо на малій відстані зору. Для основного тексту рекомендується встановлювати інтерліньяж приблизно в 1.6–1.8 разів більше від розміру шрифту. Такий добір збільшує відстань між рядками та запобігає «зімкненню» тексту, знижуючи втому очей. Для заголовків можна застосовувати менший коефіцієнт (близько 1.2–1.4), особливо якщо заголовок займає кілька рядків: це візуально відділяє його від основного тексту.

Для читабельності текст повинен добре контрастувати з фоном. Відповідно до рекомендацій WCAG, для звичайного тексту коефіцієнт контрастності має бути щонайменше 4.5:1 (для великих заголовків – не менше 3:1) [49]. За недостатньої контрастності текст виглядає блякло та його важко розрізняти, особливо при яскравому освітленні екрана. Для перевірки контрасту можна використовувати спеціальні інструменти (наприклад, WebAIM Contrast Checker). При розробці мобільних сайтів слід уникати наприклад білого тексту на світлих

фонограм чи хаотичного зображення за текстом, що знижує контраст та ускладнює читання.

Оптимальне поєднання шрифтів, розмірів, інтервалів та контрасту робить текст читабельним на малих екранах і покращує сприйняття інформації. Разом це сприяє зручному та швидкому пошуку інформації на мобільному сайті, зменшує навантаження на зір і підвищує відвідуваність. Таким чином, дотримання стандартів типографіки є важливою складовою у створенні ефективних та дружніх до користувача мобільних веб-сайтів.

2.4. Аналіз стандартів Google: Mobile-Friendly Test, Lighthouse, PageSpeed

Google приділяє особливу увагу мобільній адаптивності та швидкодії сторінок. Для перевірки цих показників існують інструменти Mobile-Friendly Test, Lighthouse та PageSpeed Insights. Інструмент Mobile-Friendly Test раніше дозволяв перевірити, чи правильно сайт відображається на смартфонах. Він оцінював наявність метатега viewport, читабельність тексту, розмір кнопок та відсутність горизонтального скролу. Google рекомендує, щоб на мобільних пристроях були доступні для натискання кнопки, зручна навігація та важлива інформація на видноті. Ті сторінки, які проходили тест Mobile-Friendly, отримували перевагу в мобільному пошуку. Навпаки, недружні до мобільних сторінки втрачають мобільний трафік. Зауважимо, що станом на грудень 2023 року інструмент Mobile-Friendly Test було виведено з експлуатації, а тепер перевірки мобільної адаптивності здійснюються в Search Console.

Lighthouse — це відкритий інструмент від Google (Chrome), який автоматично виконує аудити веб-сторінок. Він аналізує продуктивність, доступність, SEO, прогресивні веб-застосунки та інші аспекти. Lighthouse запускає сторінку у емуляції браузера і видає звіт з показниками продуктивності (наприклад, First Contentful Paint, Largest Contentful Paint, Time to Interactive) та рекомендаціями щодо виправлення помилок.

Серед ключових метрик Lighthouse – Total Blocking Time (TBT), Largest Contentful Paint (LCP), Cumulative Layout Shift (CLS), First Input Delay (FID). Наприклад, TBT вимірює загальний час блокування сторінки під час взаємодії користувача (сума частин всіх «довгих» завдань понад 50 мс), а FID оцінює затримку реакції сторінки на першу дію користувача (час від кліку до початку обробки). Lighthouse можна запускати в Chrome DevTools, у вигляді командного рядка або інтегрувати в CI/CD, тому він корисний для розробників при контролі якості сайтів.

PageSpeed Insights (PSI) — онлайн-сервіс Google для оцінки швидкодії сайту на мобільних та десктопних пристроях. Починаючи з версії 5, PSI використовує платформу Lighthouse для лабораторного аналізу й додає реальні дані користувацького досвіду (CrUX) [20]. Звіт PSI містить дві частини: Lab Data, де сторінка ему люється на віртуальному мобільному пристрої (вимірюються метрики як First Contentful Paint (FCP), Time to Interactive і формується бал від 0 до 100), та Field Data, де відображаються реальні показники з Chrome UX Report (наприклад, *FCP* та *FID* за останні 28 днів). Крім того, PSI пропонує «Можливості» з рекомендаціями щодо оптимізації та розділ «Діагностика» для додаткової інформації. Оцінка понад 90 вважається хорошою. Таким чином, PageSpeed Insights допомагає виявляти вузькі місця швидкодії та отримувати конкретні поради з оптимізації [20].

Google визначає Core Web Vitals – набір метрик, що вимірюють якість взаємодії з користувачем: швидкість завантаження, інтерактивність і стабільність контенту. Largest Contentful Paint (LCP) показує час завантаження найбільшого видимого елементу. Cumulative Layout Shift (CLS) оцінює візуальну стабільність сторінки. First Input Delay (FID) – це час від першої взаємодії користувача, до початку реакції браузера. Цей показник вимірює чутливість інтерфейсу. У лабораторних умовах Lighthouse вимірює максимальний потенціал *FID*, що відбиває найбільший можливий input-затримку для заданої сторінки. Total Blocking Time (TBT) сумує всі затримки понад 50 мс між FCP і Time to Interactive, показуючи скільки часу сторінка не відповідала на дії користувача. Всі ці

метрики напряму впливають на швидкодію та UX, тому Google радить домагатися їх оптимальних значень.

Між цими інструментами тісний зв'язок. Lighthouse — це платформа, яка лежить в основі PageSpeed Insights. PSI використовує Lighthouse для емуляції завантаження й генерує звіт з такими ж показниками та рекомендаціями. Таким чином, показники LCP, TBT, CLS, FID Lighthouse і PageSpeed є узгодженими. З іншого боку, Mobile-Friendly Test (і тепер Mobile Usability в Search Console) орієнтований на адаптивність та UI/UX на мобільних пристроях, але не вимірює перформанс напряму. Усі ці інструменти слугують єдиним стандартом «page experience». Google рекомендує спершу провести аудит у Search Console (Core Web Vitals report), а потім використовувати PageSpeed Insights і Lighthouse для детального налагодження проблем [21]. Таким чином, інструменти доповнюють один одного та надають розробникам і SEO-фахівцям повний набір даних для оцінки якості сайту.

На ранжування в пошуку Google безпосередньо впливають показники мобільності та швидкості. Мобільна адаптивність була впроваджена як фактор ранжування в мобільному пошуку з 2015 року – тепер сторінки з читабельним текстом і зручними кнопками отримують перевагу. Аналогічно, з травня 2021 року Google включив Page Experience у свій алгоритм, який поєднує Core Web Vitals [21]. Важливими факторами є мобільна дружність і безпека HTTPS. Отже, прискорення завантаження сторінок (покращення LCP, TBT, FID) та усунення зсувів (CLS) може підвищити видимість сайту у результатах пошуку. Навпаки, сайт зі слабкою мобільною оптимізацією або повільним відгуком буде знаходитись нижче у пошуку.

Для розробників ці інструменти дають чітку дорожню карту поліпшення. Lighthouse і PageSpeed Insights одразу вказують, які ресурси сповільнюють сайт, де потрібна оптимізація зображень чи скриптів, та надають конкретні рекомендації з посиланням на документацію. Це дозволяє ефективно підвищувати швидкість, мінімізувати «довгі задачі» у JavaScript та зробити UI стабільнішим. SEO-фахівці використовують звіти цих інструментів для

моніторингу відповідності сайту стандартам Google. Наприклад, регулярні перевірки з PageSpeed Insights допомагають відстежувати Core Web Vitals і готуватися до ранжування, а модуль Lighthouse SEO нагадує про базові вимоги (наявність мета-тегів, відсутність помилок пошукового індексування тощо). В комплексі, оптимізація за стандартами Google покращує користувацький досвід і водночас підтримує хороші позиції в пошуку.

РОЗДІЛ 3. ПРОЄКТНО-ПРИКЛАДНА ЧАСТИНА: РОЗРОБКА АДАПТИВНОГО ВЕБСАЙТУ

3.1. Постановка задачі та вимоги до сайту

Метою створення вебресурсу «TurboVive» є забезпечення ефективної комунікації з потенційними клієнтами, формування довіри до бренду та створення позитивного іміджу компанії. Основна задача сайту полягає в тому, щоб у найкоротші терміни донести до відвідувача ключову інформацію про сервіс, його переваги та унікальні технічні можливості «TurboVive». Для цього головний екран повинен містити чітке позиціонування – коротку тезу про надійність і точність ремонту, супроводжену привабливим візуальним контентом. Візуальний ряд має базуватися на високоякісних фонових зображеннях, які демонструють процес відновлення турбін, та підтримувати загальну колористичну концепцію компанії, що складається з відтінків синьо-сталевого спектра. Вибір типографіки орієнтований на поєднання читабельності та естетики: використання сучасних шрифтів з чіткими контурами гарантує комфортне сприйняття тексту на будь-якому пристрої.

Контактна інформація, включаючи телефон, адресу та електронну пошту, повинна бути доступна без додаткових дій користувача, тому передбачено фіксований блок із контактами внизу сторінки та можливість виклику відповідного розділу через «кнопку зв'язку» у шапці. Це дозволить відвідувачам негайно ініціювати звернення до сервісу, що є критично важливим у випадку термінових замовлень.

3.2. Проєктування структури та дизайну мобільного інтерфейсу

Структура «TurboVive» має бути розроблена з урахуванням адаптивності. У проєкті буде окремий CSS-файл `adaptive.css`, де за допомогою `@media`-правил будуть описані налаштування стилів для різних діапазонів ширини екрана. HTML структура буде розкладена на секції (`header`, `hero`, контент, `футер`), а в файлі `adaptive.css` будуть використані медіа-запити, щоб при перегляді на смартфоні

деякі блоки займали всю ширину, а вміст при цьому залишався у читабельному форматі. Скрипт `script.js` буде відповідати за інтерактивність і бургер-меню. Таким чином, архітектура «TurboVive» забезпечить мобільний досвід завдяки поєднанню адаптивного CSS та чіткої блокової структури.

Інтерфейс смартфона повинен сприйматися максимально інтуїтивно: елементи потрібно розташувати так, щоб користувач відразу зрозумів, де натискати. Як зазначено у професійних рекомендаціях, інтерфейс має бути максимально інтуїтивним та зрозумілим з чіткою логічною структурою [47]. У «TurboVive» це буде досягнуто через звичне розташування кнопок (бургер-меню у верхньому лівому куті, логотип – по центру, заклик до дії – виділеною кнопкою), зрозумілі підказки і великі області дотику. Візуальні анімації (наприклад, плавна поява меню або зміна стану кнопки при натисканні) додадуть реактивності й підсилюють враження від взаємодії.

3.3. Вибір середовища розробки та технологій

Як середовище розробки для сайту «TurboVive» обрано Visual Studio Code (VS Code). Це легкий, кросплатформний редактор з відкритим кодом, який можна гнучко налаштовувати під потреби проєкту. VS Code має розгалужену екосистему плагінів для підтримки нових мов та інструментів. Він безкоштовний і сумісний із різними операційними системами. Додатково VS Code підтримує численні мови програмування та розмітки, що робить його універсальним рішенням для веб-розробки. Також редактор містить інтегрований засіб відладки, що спрощує тестування коду [53].

Вбудований термінал, Git-інтеграція та підсвічування синтаксису в VS Code скоротили час розробки. Прозорість HTML/CSS/JS-коду полегшила внесення змін та налагодження. Крім того, модульна структура коду дозволяє швидко додавати нові функції без переробки всього сайту. У результаті всі задачі проєкту виконано швидко й якісно: сайт адаптивний, швидко завантажується і легко підтримується.

VS Code має велику екосистему розширень і налаштовуваних тем. Інтуїтивний інтерфейс та автодоповнення полегшують набір коду, а підсвічування синтаксису миттєво виділяє ключові елементи, що підвищує читабельність. Інтегрований термінал дозволяє виконувати shell-команди (наприклад, git) без виходу з редактора, а вбудована підтримка Git забезпечує зручне відстеження змін. Також VS Code споживає мало ресурсів і швидко запускається, що забезпечує комфортну роботу навіть на слабших машинах [52–55].

Незважаючи на переваги, у VS Code є недоліки. Він має обмежену функціональність порівняно з повноцінними IDE. Для веб-розробки відсутній власний візуальний рендер: щоб переглянути HTML/CSS, потрібно запускати браузер або встановлювати додаткові розширення (наприклад, Live Server). Багато корисних функцій доступні тільки через плагіни (лінтери, форматери тощо), тому необхідність додаткових інсталяцій може ускладнити налаштування. На слабких машинах велика кількість одночасних розширень може знижувати швидкодію.

У виборі технологій обрано стек HTML5, CSS3 та JavaScript (без фреймворків). HTML5 забезпечує семантичну розмітку, що структурує вміст сторінки. CSS3 додає сучасні засоби стилізації, включаючи підтримку гнучких макетів (flexbox, grid) та анімацій, що полегшує оформлення інтерфейсу. Стили розділено на main.css (базові правила) і adaptive.css (адаптивні налаштування), що спрощує організацію коду. JavaScript без фреймворків вибрано через невеликий масштаб сайту.

Такий стек оптимальний для невеликого адаптивного сайту. Відсутність важких бібліотек забезпечує швидке завантаження та спрощує код. Крім того, застосування стандартних технологій полегшує залучення до роботи інших розробників. Для малих проєктів «складність і накладні витрати фреймворку можуть перевищувати переваги», а ванильний JavaScript забезпечує більш ефективне рішення [3]. Семантичні теги HTML5 також покращують SEO та доступність сайту.

3.4. Реалізація адаптивного макету: HTML, CSS, JS

Мова гіпертекстової розмітки HTML

Сайт TurboVive побудований за модульним принципом: кожен розділ позначено окремим блоком. У верхній частині знаходиться header зі статичною навігацією. `<div class="header">` містить кнопки навігації і лого. Всередині блоку присутні фіксована панель меню та бургер-кнопка для мобільного меню. Далі під шапкою розташовані основні розділи сайту. Блок «Наші цінності» позначено `<div id="service" class="service__block">` який містить заголовок та підрозділи з описом цінностей. Секція FAQ розташована у блоці `<div id="gallery" class="gallery__block">` і складається із заголовка та переліку статей «Запитання-відповідь». Також є блок «Чому ми» (`id="about-me"`) з описом компанії та ілюстрацією, а наприкінці – блок «Контакти» (`<div id="contacts" class="contacts__block">`) з адресою, email та графіком. Футер реалізований тегом `<footer>` із стандартною копірайт-рядком (*Див. Додаток А (HTML файл)*).

Базове CSS-оформлення

Файл `main.css` задає стилі для базового оформлення. У ньому контейнер `.main__container` визначено як CSS Grid з трьома рівними колонками (`grid-template-columns: repeat(3, 1fr)`), що дозволяє гнучко розташувати основні секції. Блок header займає всю ширину (`grid-column: 1/4`) і задає фон зображення турбіни (`background: url('./Img__background/turbo.png')`) з фіксованим прикріпленням. Для навігації `.nav` використано Flexbox, вирівнювання по центру і темно-синє тло (`background-color: rgb(101, 119, 163)`). Пункти меню `.nav__items` стилізовані світлим текстом і роблять колір світлішим при наведенні (`hover`). Заголовки секцій (клас `section-title`) вирівняні по центру, приведені до верхнього регістру та мають великі розміри шрифту (32px, напівжирний).

Типографіка задається через підключені шрифти Google Fonts (Itim, Ubuntu, PT Sans) і стилі для тексту. Наприклад, абзац у header має ширину 500px і шрифт 21. Кольори тексту – в основному темно-сірі або чорні, що контрастує з фоном. Соціальні іконки у шапці оформлені як круглі з напівпрозорим фоновим кольором і змінюють фон на білий при наведенні.

Блок `.header__main` – контейнер з двома колонками (зображення і текст) – має `display: flex` і `flex-direction: row`. Секція цінностей `.service__block` теж застосовує `display: flex`; `flex-wrap: wrap`; `justify-content: center` для розташування трьох підблоків поруч із вирівнюванням по центру. Загалом CSS відповідає за відступи, розміри і фон секцій, забезпечуючи єдиний стиль макету (наприклад, фонові прозорість і округлені кути у `.service__block`). Використання модулю Flexbox дає можливість розподілу простору між елементами інтерфейсу при зміні ширини екрану (*Додаток Б - CSS файл, базові стилі*). Цей підхід не змінює базову структуру сайту і не потребує змін або додавання додаткових елементів у HTML документ. Таким чином адаптація макету під різні пристрої навантажує функціонал і не потребує часу для завантаження додаткових елементів сайту.

Детальніше розглянемо стратегію адаптивності через CSS. Адаптивні стилі винесені в окремий файл `adaptive.css`, який містить медіа-запити для різних ширин екрану. Для мобільних пристроїв (ширина до 480px) вказано `@media (max-width: 480px)`: у цьому випадку головний заголовок `.header__main` переходить у вертикальне розташування (`flex-direction: column`), картинка і текст центруються (видалено ліві відступи, ширина тексту стає 100% або 79% ширини контейнера). Текст і кнопки центровані за рахунок `text-align: center`, а шрифт трохи зменшено (для `.header__text` `p font-size 18px`). Секція цінностей `.service__block` отримує автоматичну висоту і спрощене вирівнювання (блоки займають по ~90% ширини), щоб вони розміщувалися один під одним. Для планшетів (481–768px) подібні зміни: меню навігації ховається (`.nav { display: none }`), відкривається місце для бургер-меню, а всі елементи також центруються.

Наприклад, зображення в `.header__img` задається шириною 320px і центроване, текст переноситься на всю ширину і центрується, а блоки цінностей мають `width: 91%` і авто-висоту. На більших екранах (769px+) навігація повертається, меню видно звичним чином, а розміри блоків набувають статичних значень (у великих екранах `.header__main { height: 500px; }`).

Особливо виділено адаптацію меню: у стилях є бургер-кнопка `.openbtn`, яка прихована за замовчуванням, але стає видимою при `@media (max-width: 768px)`. Одночасно у вузьких екранів `.nav` не показується, а на маленьких – бокова панель меню займає весь екран (від 0% до 100% ширини). Таким чином, за допомогою медіа-запитів забезпечується плавна перебудова макету. Колонки стають рядами, меню адаптується, а текстові блоки центруються.

Інтерактивність виконана за допомогою JavaScript (`script.js`). Файл `script.js` додає необхідну динаміку. Для реалізації анімацій при скролі скрипт збирає всі елементи з класом `.js-scroll` і перевіряє їхню позицію у вікні. Якщо елемент опиняється в полі зору (`elementInView`), до нього додається клас `scrolled`, що через CSS-ключові кадри запускає анімацію (`fade-in`, `slide-left` тощо). Коли елемент виходить з області перегляду, клас прибирається. У результаті елементи плавно з'являються або зсуваються під час прокрутки.

Нарешті, скрипт забезпечує роботу бургер-меню: при натисканні на кнопку з `id="burgerBtn"` викликається функція `openNav()`, яка робить бічну панель меню видимою (`style.width = "100%"`) і приховує саму кнопку. Кнопка-закриття (`id="closeBtn"`) зменшує ширину панелі до 0%, відновлюючи вихідне оформлення і повертаючи видимість бургеру. Крім того, кожне посилання всередині панелі навігації при натисканні також закриває панель. Таким чином, JavaScript додає основні взаємодії: анімацію прокрутки і динамічне меню, не потребуючи сторонніх бібліотек (*Додаток Г - Java Script*).

Таким чином, описаний підхід відповідає принципам сучасної верстки: застосування семантичних блоків, модульність і розділення стилів (окремі файли для базових стилів і адаптивних налаштувань). Використання Flexbox та CSS Grid спрощує макетування і робить розмітку гнучкою, а медіа-запити – зрозумілим інструментом для адаптації. Поділ стилів на `main.css` та `adaptive.css` полегшує підтримку коду: базові стилі відповідають за загальний вигляд, а `adaptive`-файл – за зміни під різні пристрої. Кодування на чистому JavaScript без фреймворків теж мінімізує вагу сторінки і складність: функціонал реалізовано лаконічно, лише для необхідних елементів.

Загалом, такий підхід поєднує простоту та ефективність. Це сприяє оптимізації продуктивності (мало зовнішніх залежностей, апаратно-акселераційні анімації) та зручному супроводу коду (логічна структура файлів і налаштувань).

3.5. Тестування сайту на різних екранах і пристроях

Адаптивний дизайн забезпечує коректне відображення сайту на різних пристроях – від смартфонів до великих моніторів. Такий підхід дозволяє користувачам отримувати однакову функціональність та зручний інтерфейс за будь-якої ширини екрану. Тестування адаптивності гарантує високу зручність, зниження відмов і загальне поліпшення користувацького досвіду [42].

Для перевірки адаптивності використовувався вбудований емулятор пристроїв у Google Chrome DevTools. Цей режим дозволяє імітувати різні екрани: він встановлює задану ширину/висоту, змінює тип пристрою та дозволяє емулювати обмеження мережі або процесора. Важливо розуміти, що це лише наближення, деякі особливості апаратного забезпечення він імітувати не може [17].

Для активації емулятора достатньо відкрити DevTools (натисканням F12 або Ctrl+Shift+I) і натиснути кнопку Toggle device toolbar у верхній панелі. За замовчуванням відкривається «Responsive» режим з довільною шириною, але у списку Dimensions можна вибрати конкретні еталонні пристрої (наприклад, iPhone, iPad, Pixel тощо). Це дає змогу швидко переключатися між моделями та відстежувати зміну макету сайту в реальному часі.

Ключові *брейкпоінти* – це значення ширини екрану, при яких застосовуються інші стилі через CSS-медіа-запити. В файлі адаптивних стилів сайту налаштовані наступні основні діапазони:

- До 480px – мобільні телефони;
- 481-768px – планшети;
- 769-1199px – ноутбуки та середні екрани;
- Від 1200px – великі екрани.

Ці пороги охоплюють типові класи пристроїв і дозволяють коригувати макет для кожного з них. Медіа-запити у CSS керують таким змінним макетом, аби він залишався «гнучким і адаптивним» для будь-яких пристроїв.

Під час тестування використовувалися два основні підходи:

Ручна зміна ширини вікна браузера. За допомогою миші плавно витягували правий край вікна, спостерігаючи перехід між медіа-запитами і відповідні зміни в розташуванні елементів. Це дозволило перевірити поведінку сайту на довільних проміжних ширинах

Емулятор пристроїв у Chrome DevTools. Послідовно активували Device Mode (F12 → клік по іконці пристрою), а потім у меню Dimensions обирали стандартні конфігурації (наприклад, iPhone X, Google Pixel 4, iPad Pro тощо). У цьому режимі сайт відкривається у вікні з фіксованою роздільною здатністю, що показує, як дизайн виглядатиме на конкретному пристрої. Обидва методи (перетягування та вибір готових пристроїв) використовувалися доповнююче для переконливого результату.

Поведінка сайту на різних екранах:

Мобільні телефони ($\leq 480px$) (Додаток В - Мобільні телефони ($\leq 480px$)).

На найменших екранах до 480 px макет максимально спрощується: шапка сайту переходить у вертикальну колонну структуру завдяки правилу @media (max-width: 480px) { .header__main { flex-direction: column; } }, логотип у блоці .header__img центрується над текстом із фіксованою шириною близько 280 px, а текстова частина .header__text займає 79 % ширини та вирівнюється по центру, при цьому розмір шрифту абзацу знижується до 18 px для кращої читабельності. Горизонтальне меню приховується (.nav { display: none; }), натомість відображається бургер-кнопка (.openbtn { display: block; }), що відкриває бокову панель із навігацією. Секція «Наші цінності» адаптується так, що заголовок розтягується на всю ширину й центрується, а окремі блоки цінностей мають ширину приблизно 90 % і також розташовуються по центру сторінки.

Планишети (481–768px) (Додаток Г - Планишети (481–768px)). При розмірах від 481 px до 768 px дизайн продовжує наслідувати мобільну логіку, але

з дещо більшими елементами: у блоці `.header__img` задається ширина 320 px, текст у заголовку займає близько 90 % ширини, а стилі вирівнювання `.header__text p`, `.header__text span { text-align: center; }` зберігають комфортне сприйняття. Пристрої в цьому діапазоні також використовують бургер-меню, оскільки навігація залишається прихованою, а кнопка для її відкриття — доступною.

Ноутбуки та середні екрани (769-1199px) (Додаток Д - Ноутбуки та середні екрани (769-1199px)). На екранах шириною від 769 px до 1199 px активується класичний десктопний макет: навігаційне меню відображається горизонтально (`.nav { display: flex; }`), шапка повертається до рядкового вигляду з логотипом зліва та текстом праворуч, при цьому `.header__img { width: 350px; }` і `.header__text { width: 50%; }`. Параграфи вирівнюються по ширині (`text-align: justify`) та збільшуються до приблизно 21 px, що відповідає зручному читанню на ноутбуках і моніторах. У секції «Наші цінності» реалізовано багатостовпкову сітку з `flex-wrap: wrap; gap: 202px;`, завдяки чому три інформаційні блоки розташовуються поруч.

Великі екрани (≥ 1200 px) (Додаток Е - Великі екрани (≥ 1200 px)). Нарешті, для екранів від 1200 px і вище визначено медіа-запит `@media (min-width: 1200px)`, у рамках якого `.main__container { max-width: 100%; margin: 0 auto; }` дозволяє використовувати повну ширину вікна перегляду, а висота шапки зменшується до 500 px для компактності. Таким чином, за допомогою чотирьох чітко окреслених брейк-поінтів забезпечується плавний перехід від лаконічного мобільного інтерфейсу до багатостовпкового десктопного макета, що гарантує оптимальне відображення контенту та зручність користування на будь-якому пристрої.

Отримана поведінка сайту повністю відповідає записаним медіа-запитам і очікуванням. При малих ширинах `.nav` прихована, а `.openbtn` видима, як визначено в `adaptive.css`; натискання на бургер відкриває меню (скрипт `openNav` і `closeNav` змінюють ширину бокової панелі на 100%). Стилi шапки (flex-колонка, центровані тексти) та розділів (повна ширина заголовків, зменшені шрифти)

застосовуються вірно і не призводять до прокрутки чи накладання контенту. JavaScript-анімації прокрутки і галереї працюють коректно на всіх ширинах. Виявлених помилок або значних розривів у макеті не було – спостережувані зміни («поворот» навігації, перехід блоків у стовпець, зміна розмірів зображень і шрифтів) суворо слідує правилам `adaptive.css` і логіці сайту. Варто зазначити, що емуляція Chrome DevTools лише наближено відображає мобільний досвід, тому остаточну перевірку бажано провести на реальних пристроях і у різних браузерах.

У цілому тестування показало, що сайт TurboVive успішно адаптований до основних класів пристроїв. Сильні сторони – чітка та передбачувана зміна макету, збереження функціональності та зручності навігації (гамбургер-меню, центрування контенту, масштабування елементів). Користувачі на телефонах отримують спрощену але повнофункціональну версію, а на ПК бачать багатоколонковий, інформативний вигляд. Виявлено лише незначні неточності: наприклад, на деяких проміжних ширинах може спостерігатися незграбність відступів або невеликі розриви в розташуванні елементів, що може потребувати тонкого налаштування CSS. Можливий напрямок покращень – введення додаткових проміжних брейкпоінтів для екзотичних роздільностей або оптимізація графіки для прискорення завантаження. Однак у даному вигляді адаптивність TurboVive повністю відповідає вимогам і забезпечує комфортний перегляд на різних пристроях.

3.6. Перевірка швидкодії та юзабіліті: результати Lighthouse

Lighthouse — це відкритий інструмент автоматизованого аналізу веб-сторінок, розроблений Google, який проводить серію аудитів для оцінювання якості сайту. Він вимірює кілька ключових показників і формує звіт із загальними балами та рекомендаціями.

В результаті аудиту TurboVive отримано такі оцінки: продуктивність – 71–74 бали, доступність – ~96 балів, SEO – ~73 бали (зі 100). Значення продуктивності ~72 вказує на середній рівень і потребу оптимізації, тоді як

висока оцінка доступності (96) свідчить про майже повне дотримання стандартів доступності. *(Додаток Є - Результати аудиту Google Lighthouse).*

Під час аудиту виявлено такі проблеми:

- Зображення не мають явних атрибутів width і height, що може спричиняти зсуви макета при завантаженні;
- Деякі файли зображень значно перевищують відтворюваний розмір, через що сторінка завантажується повільніше;
- Відсутня структурована мікророзмітка SEO (наприклад, schema.org), через що сторінка не розкриває додаткових даних пошуковим системам.

До позитивних сторін можемо віднести:

- швидке завантаження: Total Blocking Time = 0 мс (гарантує відсутність довгих завдань і миттєву взаємодію);
- стабільність макета: Cumulative Layout Shift = 0 (немає несподіваних зсувів елементів, що забезпечує плавний UX);
- висока доступність: оцінка ~96 балів означає, що майже всі критичні аудити доступності пройдено успішно.

Сайт TurboVive продемонстрував сильні сторони у швидкості завантаження та високій доступності, але виявлені проблеми в оптимізації зображень та базових SEO-настановах (наприклад, відсутність мета-опису).

Для покращення продуктивності та SEO рекомендується:

- Оптимізувати зображення: стиснути та подавати їх у відповідному розмірі;
- Задавати явні атрибути width і height для всіх , щоб резервувати місце і уникнути CLS;
- Впровадити структуровану мікророзмітку (schema.org) для важливих даних (товари, статті, контактну інформацію тощо).

Впровадження цих рекомендацій прискорить завантаження, покращить досвід користувачів і підвищить позиції сайту в пошуковій видачі.

3.7. Пропозиції щодо вдосконалення розробленого рішення

На основі детального аналізу поточної реалізації сайту TurboVive (HTML-структури, CSS-стилів, JS-логіки) та результатів адаптивного й Lighthouse-тестів сформульовано низку рекомендацій щодо поліпшення. По-перше, слід використовувати семантичні HTML5-елементи (наприклад, `<nav>`, `<header>`, `<main>`, `<section>`, `<footer>`), а не суто `<div>`-контейнери. Це створить чітку ієрархію контенту: наприклад, `<nav>` для меню, `<main>` – для основного вмісту, `<footer>` – для футера. Такий підхід робить розмітку зрозумілою як для браузерів і пошукових систем, так і для допоміжних технологій – екранні рідери можуть однозначно ідентифікувати блоки інформації. Також необхідно правильно розставити заголовки `<h1>`-`<h6>` у логічній послідовності та додати атрибут `lang="uk"` (замість `lang="en"`), оскільки вміст сторінки українською. Коректна семантика HTML дає значні переваги: вона важлива для доступності і пошукової оптимізації і забезпечує зручну навігацію для користувачів з інвалідністю (наприклад, правильне використання `<nav>`, `<main>`, `<footer>` сприяє передбачуваній навігації за допоміжними засобами) [1].

Стосовно оптимізації HTML/CSS-коду, то код слід зробити більш модульним і структурованим. Рекомендується застосувати методологію найменувань класів (наприклад, BEM) для кращого розуміння і повторного використання компонентів. Використання незалежних блоків у стилях знижує зв'язність і полегшує підтримку: незалежні «блоки» можна легко переносити й комбінувати, що зменшує обсяг CSS-коду і покращує підтримуваність. Імена класів мають бути зрозумілими (замість `first__icon1`, `first__icon2` краще `icon-honesty`, `icon-technical` тощо), що полегшить орієнтування в коді. У стилях доцільно об'єднати схожі властивості, замінити фіксовані значення ширини на відносні (`%`, `rem`), а також видалити невикористані селектори.

Особливу увагу слід приділити роботі з зображеннями: кожному `<img>` необхідно задати атрибут `alt` з описом – це не лише підвищить доступність (екранні рідери озвучать зміст зображення), але й сприяє SEO. Зображення варто оптимізувати (зжати і за можливості переконвертувати у сучасні формати,

наприклад WebP) та вказати атрибути width/height для запобігання зсувам макета. Також можна реалізувати «ліниве» завантаження (`loading="lazy"`) і адаптивні зображення (`srcset`) для економії трафіку та пришвидшення першого рендеру [51]. Окрім того, слід додати необхідні мета-теги (наприклад, `<meta name="description">`), і забезпечити коректне підключення зовнішніх ресурсів. Наприклад, Google Fonts можна підвантажувати з використанням `rel="preload"` і правила `font-display`, щоб уникнути порожніх інтервалів при завантаженні шрифтів.

Адаптивні стилі слід доповнити додатковими точками перелому для більш плавних змін макета на різних пристроях. Наразі охоплено базові розміри (мобільні, планшети, ноутбуки, десктопи), але можна врахувати також середні десктопні монітори та великі екрани. Рекомендується застосовувати принцип `mobile-first`: стилі з мінімальною шириною (наприклад, `min-width: 320px`) розширювати до більших. Завдяки цьому інтерфейс збережеться читабельним на всіх розмірах: наприклад, меню стане доступнішим, а контент – рівномірно розставленим без різких «стрибків» стилів при зміні орієнтації чи розміру вікна браузера.

JavaScript-код слід зробити більш модульним і ефективним. Для обробників подій доцільно використовувати делегування: замість додавання слухача події до кожного елемента галереї за іменем (`document.getElementsByName('image')`), можна повісити один обробник на контейнер і визначати ціль через `event.target`. Такий підхід дозволяє опрацювати велику кількість дочірніх елементів одним обробником, що спрощує код і зменшує навантаження. Аналогічно, для бургер-кнопки та інших елементів краще застосовувати `addEventListener` із коректними селекторами. Взагалі, сучасні селектори DOM (`querySelector/querySelectorAll`) є гнучкішими та зручнішими за застарілі методи типу `getElementsByName`, які зараз рідко застосовують. Наприклад, можна призначати елементи за унікальними ідентифікаторами або `data-` атрибутами і шукати їх через `document.querySelector('#myElement')`. Крім того, варто вилучити закоментований

або невикористаний код (наприклад, незакінчена валідація форми), оскільки порожні функції лише ускладнюють підтримку.

Для підвищення продуктивності та SEO слід впровадити низку сучасних практик. По-перше, мікророзмітка: додавання schema.org тегів (наприклад, тип LocalBusiness для компанії та ContactPage для контактів) допоможе пошуковим системам краще розуміти зміст сторінки. Крім того, варто налаштувати дедалі більші кеші на сервері. Статичним ресурсам (скрипти, стилі, зображення) рекомендується задавати Cache-Control з тривалим TTL (наприклад, max-age=31536000), щоб браузері могли зберігати їх протягом року без повторних запитів. Це значно пришвидшить повторні завантаження сторінок та знизить навантаження на сервер.

По-друге, оптимізувати підключення шрифтів: вже використані preconnect/preload-підказки виправдані, але слід також встановити font-display: swap або optional у @font-face, щоб уникнути «пробілів» або зсувів контенту, поки завантажуються Web-шрифти. Рекомендується перенести блоки <script> перед закриттям </body> або додати атрибут defer, щоби скрипти не блокували первинний рендер. Також корисно згорнути та мінімізувати CSS/JS файли, а при великих об'ємах підключити їх через CDN. В сукупності ці практики (структуровані мета-теги, мікророзмітка, оптимізація шрифтів, кешування) гарантують кращі показники продуктивності та SEO згідно з рекомендаціями Lighthouse та Google.

І на останок, сайт має стати доступнішим для всіх користувачів. Варто перевірити й поліпшити контрастність кольорів (згідно з WCAG-стандартом – не менше 4.5:1 для основного тексту), а також переконатися, що всі клікабельні елементи мають достатню площу та доступні з клавіатури. З об'єктами слід працювати з урахуванням ARIA: наприклад, кнопки «бургер-меню» додати aria-label="Відкрити меню" та змінювати aria-expanded при відкритті/закритті, щоб користувачі скрінрідерів розуміли її стан. Семантичні блоки <nav>, <header>, <main>, <footer> уже розставлені за рекомендаціями: їх використання «становить більш передбачувану структуру» і покращує навігацію для допоміжних

технологій. Також усі значимі зображення потрібно забезпечити описовим alt (екрани читатимуть цей текст). Якщо на сайті з'являться форми зв'язку, їх поля треба пов'язати з <label>, а порядок елементів у DOM має відповідати логіці взаємодії. У сукупності ці поліпшення зроблять взаємодію з сайтом зручнішою і безпечнішою для людей з обмеженими можливостями або незвичними пристроями доступу.

Впровадження перерахованих удосконалень значно підвищить загальну якість рішення. Семантична розмітка та коректні заголовки сприятимуть кращій індексації сайту й доступності (HTML-структура повинна передавати зміст у зрозумілому вигляді), а оптимізація зображень і кешування – швидшому завантаженню сторінок. Використання методології на кшталт BEM зробить CSS більш модульним і підтримуваним, що прискорить подальшу розробку. Разом із застосуванням сучасних практик (асинхронне підключення ресурсів, мікророзмітка, ARIA-атрибути) це забезпечить сумісність зі стандартами W3C і WCAG, позитивно позначиться на User Experience і критеріях Lighthouse. Таким чином, реалізація цих пропозицій підвищить продуктивність сайту, полегшить його супровід і зробить рішення більш відповідним сучасним веб-стандартам.

ВИСНОВКИ

У роботі було здійснено комплексний аналіз теоретичних основ і практичну реалізацію адаптивного веб-ресурсу TurboVive з урахуванням сучасних вимог до мобільного та десктопного UX/UI. Далі наведемо підсумки, у яких узагальнимо отримані результати, зробимо висновки та окреслимо напрямки подальшого розвитку та удосконалення розробленого рішення.

Перш за все, теоретична частина дозволила обґрунтувати необхідність використання адаптивного та респонсивного дизайну в умовах стрімкого росту мобільного трафіку. Аналіз світових тенденцій показав, що частка мобільних пристроїв у загальному обсязі користувацьких звернень перевищує 60 %, і надалі буде зростати. Відповідно було досліджено еволюцію користування інтернетом, класифікацію мобільних пристроїв за розмірами та роздільною здатністю екранів, а також теоретичні підходи до побудови адаптивних макетів. Окремо було розглянуто базові принципи UX/UI для мобільних інтерфейсів, серед яких — принцип “content-first”, mobile-first strategy, забезпечення читабельності тексту, достатні області дотику та інтуїтивні патерни навігації (бургер-меню, фіксовані СТА). Ці знання стали основою для побудови практичної моделі сайту TurboVive.

У практичній частині було виконано розробку макета та структурування контенту на основі HTML5. Використання логічних тегів не лише покращило читабельність коду, а й підвищило доступність ресурсу для користувачів із обмеженими можливостями та сприяло коректній індексації пошуковими системами. Для стилізації було застосовано CSS3. Flexbox забезпечив гнучке компонування елементів у шапці та сервісних блоках, а CSS Grid — адаптивну сітку блоку статей. Базові стилі винесено до файлу main.css, а адаптивні налаштування — до adaptive.css із трьома ключовими брейкпоінтами (≤ 480 px, 481–768 px, ≥ 769 px). Така модульна організація стилів забезпечила зручність супроводу та подальшого масштабування проєкту.

Тестування адаптивності з допомогою Chrome DevTools підтвердило коректну поведінку макета на різних розмірах вікна: вертикальна трансформація шапки та сервісних блоків на мобільних екранах, відображення бургер-меню, центровані тексти та масштабовані зображення. Аудит Google Lighthouse показав середній рівень продуктивності (71–74 бали) — через відсутність явних атрибутів width/height у зображеннях та необхідність оптимізації графіки, але високі показники доступності (~96 балів) і стабільності макета (CLS = 0) підтвердили успішність реалізованих рішень. Бали SEO (~73) стали підставою для розробки рекомендацій з впровадження мікророзмітки schema.org та оптимізації мета-тегів

Таким чином, виконане дослідження підтвердило важливість комплексного підходу до адаптивної розробки та окреслило план подальшого розвитку вебресурсу TurboVive відповідно до сучасних стандартів і потреб аудиторії.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. AccessiblyApp. “Semantic HTML”. URL: <https://accessiblyapp.com/blog/semantic-html/> (дата звернення: 16.05.2025)
2. Adidas Україна. “Adidas Україна”. URL: https://www.adidas.ua/?srsltid=AfmBOooz6Vjxcop_qpt-KnroTN5uBdXh8LoC87pJUnWgUnMNCc5XZugi (дата звернення: 16.05.2025)
3. AST Consulting. “JavaScript without Frameworks: Back to Basics”. URL: <https://astconsulting.in/java-script/javascript-without-frameworks-back-to-basics> (дата звернення: 16.05.2025)
4. Backlinko Team. “Smartphone Usage Statistics”. Останнє оновлення 30 січня 2025 р. URL: <https://backlinko.com/smartphone-usage-statistics> (дата звернення: 16.05.2025)
5. Bob Sparkins. “Mobile Landing Page Examples: See High-Converting Pages, Best Practices, Templates & More”. Опубліковано 1 грудня 2023 р.; оновлено 13 травня 2025 р. URL: <https://www.leadpages.com/blog/mobile-landing-page-examples> (дата звернення: 16.05.2025)
6. Charles Matthew Atos. “Mastering Mobile Typography: Font Usage Tips and Best Practices”. Toptal. URL: <https://www.toptal.com/designers/typography/typography-for-mobile-apps> (дата звернення: 16.05.2025)
7. CompassAI. Mobile SEO Best Practices 2025. 25 лютого 2025 р. URL: <https://www.upskillist.com/blog/mobile-seo-best-practices-2025/> (дата звернення: 16.05.2025)
8. Cube Creative. “Top Social Media Statistics: Insights for Small Businesses”. 25 вересня 2023 р. URL: <https://cubecreative.design/blog/small-business-marketing/from-zero-to-hero-social-media-stats-to-transform-your-small-business> (дата звернення: 16.05.2025)
9. CSS-Tricks. “Viewport-Sized Typography”. URL: <https://css-tricks.com/viewport-sized-typography> (дата звернення: 16.05.2025)

10. DataReportal. Digital Around the World. URL: <https://datareportal.com/global-digital-overview> (дата зверення: 16.05.2025)
11. Dev.to. “Responsive Typography: Making Text Adapt to All Screens”. URL: <https://dev.to/magsimuhammadessa/responsive-typography-making-text-adapt-to-all-screens-29p3> (дата зверення: 16.05.2025)
12. DXOMARK. “Sony Xperia 1 IV Display test”. 29 листопада 2022 р. URL: <https://www.dxomark.com/sony-xperia-1-iv-display-test/> (дата зверення: 16.05.2025)
13. D Dhruv. “How Do Mobile-First Designs Influence SEO Rankings?”. Друкарня. 15 квітня 2025 р. URL: <https://drukarnia.com.ua/articles/how-do-mobile-first-designs-influence-seo-rankings-tBrPE> (дата зверення: 16.05.2025)
14. Друкарня. “Responsive чи adaptive дизайн”. URL: <https://drukarnia.com.ua/articles/responsive-chi-adaptive-dizain-KAnOA> (дата зверення: 16.05.2025)
15. FlatpanelsHD. URL: <https://www.flatpanelshd.com/news.php?subaction=showfull&id=1719291889> (дата зверення: 16.05.2025)
16. GeeksforGeeks. “Basics of Responsive Web Design: Media Queries”. URL: <https://www.geeksforgeeks.org/basics-of-responsive-web-design-media-queries/> (дата зверення: 16.05.2025)
17. Google Developers. “DevTools Device Mode”. URL: <https://developer.chrome.com/docs/devtools/device-mode> (дата зверення: 16.05.2025)
18. Google Developers. “Lighthouse Overview”. URL: <https://developer.chrome.com/docs/lighthouse/overview> (дата зверення: 16.05.2025)
19. Google Developers. “Lighthouse Total Blocking Time”. URL: <https://developer.chrome.com/docs/lighthouse/performance/lighthouse-total-blocking-time> (дата зверення: 16.05.2025)
20. Google Developers. “PageSpeed Insights, now powered by Lighthouse”. Google Search Central Blog. 12 листопада 2018 р. URL:

<https://developers.google.com/search/blog/2018/11/pagespeed-insights-now-powered-by> (дата зверення: 16.05.2025)

21. Google Developers. “Timing for bringing page experience to Google Search”. Google Search Central Blog. 10 листопада 2020 р. URL: <https://developers.google.com/search/blog/2020/11/timing-for-page-experience> (дата зверення: 16.05.2025)

22. Google Fonts. “Roboto”. URL: <https://fonts.google.com/specimen/Roboto> (дата зверення: 16.05.2025)

23. Hypersense Software. “The Business Impact of User Experience (UX): How User Experience Drives Revenue and Reduces Costs”. 17 травня 2024 р. URL: <https://hypersense-software.com/blog/2024/05/17/business-impact-ux/> (дата зверення: 16.05.2025)

24. Jake Archibald. “Caching Best Practices”. URL: <https://jakearchibald.com/2016/caching-best-practices/> (дата зверення: 16.05.2025)

25. Justinmind. “Hamburger Menu”. URL: <https://www.justinmind.com/ui-design/hamburger-menu> (дата зверення: 16.05.2025)

26. McDonald’s. “McDonald’s Україна”. URL: <https://www.mcdonalds.com/ua/uk-ua.html> (дата зверення: 16.05.2025)

27. Mozilla Developer Network. “Media queries”. URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/CSS_layout/Media_queries (дата зверення: 16.05.2025)

28. Mozilla Developer Network. “Multimedia Best Practices for Web Performance”. URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Performance/Multimedia (дата зверення: 16.05.2025)

29. Mozilla Developer Network. “Progressive Enhancement”. URL: https://developer.mozilla.org/en-US/docs/Glossary/Progressive_Enhancement (дата зверення: 16.05.2025)

30. Mozilla Developer Network. “Using media queries”. URL: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_media_queries/Using_media_queries (дата зверення: 16.05.2025)
31. Mozilla Developer Network. “Values and Units”. URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Styling_basics/Values_and_units (дата зверення: 16.05.2025)
32. Mozilla Developer Network. “Viewport <meta> element”. URL: https://developer.mozilla.org/en-US/docs/Web/HTML/Guides/Viewport_meta_element (дата зверення: 16.05.2025)
33. Nielsen Norman Group. “Basic Patterns for Mobile Navigation: A Primer”. 15 листопада 2015 р. URL: <https://www.nngroup.com/articles/mobile-navigation-patterns/> (дата зверення: 16.05.2025)
34. Nielsen Norman Group. “Visual Hierarchy in UX: Definition”. URL: <https://www.nngroup.com/articles/visual-hierarchy-ux-definition/> (дата зверення: 16.05.2025)
35. Ozdoruk C. “75 Jaw-Dropping Conversion Rate (CRO) Statistics You Need in 2024.” Firework. URL: <https://firework.com/blog/conversion-rate-optimization-statistics> (дата зверення: 16.05.2025)
36. Omniconvert. “Above-the-Fold Design”. URL: <https://www.omniconvert.com/blog/above-the-fold-design/> (дата зверення: 16.05.2025)
37. PixelFreeStudio. “How to Use Lighthouse to Audit Your PWA”. URL: <https://blog.pixelfreestudio.com/how-to-use-lighthouse-to-audit-your-pwa/> (дата зверення: 16.05.2025)
38. Reuters. “False claim: 3G, 4G and 5G caused SARS, swine flu and the new coronavirus”. Reuters Fact Check. 7 квітня 2020 р. URL: <https://www.reuters.com/article/world/false-claim-3g-4g-and-5g-caused-sars-swine-flu-and-the-new-coronavirus-idUSKBN21P275/> (дата зверення: 16.05.2025)

39. Romankoff. “Мікророзмітка”. URL: <https://romankoff.com/blog/seo/microrozmitka> (дата зверення: 16.05.2025)
40. Sitechecker.pro. “What Is the viewport meta tag?”. URL: <https://sitechecker.pro/what-is-viewport-meta-tag/> (дата зверення: 16.05.2025)
41. SitePoint. “An Introduction to Mobile-First Media Queries”. URL: <https://www.sitepoint.com/introduction-mobile-first-media-queries/> (дата зверення: 16.05.2025)
42. Svitla Team. “Responsive Web Design Testing”. 18 квітня 2019 р. URL: <https://svitla.com/blog/responsive-web-design-testing/> (дата зверення: 16.05.2025)
43. Sweor. “27 Eye-Opening Website Statistics: Is Your Website Costing You Clients?” Оновлено 6 березня 2023 р. URL: <https://www.sweor.com/firstimpressions> (дата зверення: 16.05.2025)
44. Unbounce. “Best Mobile Landing Page Examples”. URL: <https://unbounce.com/landing-page-examples/best-mobile-landing-page-examples/> (дата зверення: 16.05.2025)
45. UXPin. “Responsive vs Adaptive Design: What’s the Best Choice for Designers?”. URL: <https://www.uxpin.com/studio/blog/responsive-vs-adaptive-design-whats-best-choice-designers/> (дата зверення: 16.05.2025)
46. Web-promo.ua. “Як маркетологу знайти спільну мову з веб-дизайнером: вивчаємо термінологію UX/UI”. URL: <https://web-promo.ua/ua/blog/kak-marketologu-najti-obshij-yazyk-s-veb-dizajnerom-izuchaem-uxui-terminologiyu/> (дата зверення: 16.05.2025)
47. Wezom. “Дизайн інтерфейсів мобільних застосунків”. URL: <https://wezom.com.ua/ua/blog/dizajn-interfejsov-mobilnyh-prilozhenij> (дата зверення: 16.05.2025)
48. W3C WAI. “Understanding Success Criterion 2.5.5: Target Size”. URL: <https://www.w3.org/WAI/WCAG21/Understanding/target-size.html> (дата зверення: 16.05.2025)

49. W3C. “Web Content Accessibility Guidelines (WCAG) 2.1 – український переклад”. URL: <https://www.w3.org/Translations/WCAG21-ua/> (дата звернення: 16.05.2025)

50. Wizeclub Education. “Що таке респонсивний дизайн”. 1 січня 2024 р. URL: <https://wizeclub.education/blog/shho-take-responsivnij-dizajn/> (дата звернення: 16.05.2025)

51. Web.dev. “Optimize Webfont Loading”. URL: <https://web.dev/articles/optimize-webfont-loading> (дата звернення: 16.05.2025)

52. Visual Studio Code API. “Syntax Highlight Guide”. URL: <https://code.visualstudio.com/api/language-extensions/syntax-highlight-guide> (дата звернення: 16.05.2025)

53. Visual Studio Code Docs. “Configure Extensions (Extension Marketplace)”. URL: <https://code.visualstudio.com/docs/configure/extensions/extension-marketplace> (дата звернення: 16.05.2025)

54. Visual Studio Code Docs. “Source Control Overview”. URL: <https://code.visualstudio.com/docs/sourcecontrol/overview> (дата звернення: 16.05.2025)

55. Visual Studio Code Docs. “Terminal Basics”. URL: <https://code.visualstudio.com/docs/terminal/basics> (дата звернення: 16.05.2025)

56. Yoast. “Image SEO – ALT Tag and Title Tag Optimization”. URL: <https://yoast.com/image-seo-alt-tag-and-title-tag-optimization/> (дата звернення: 16.05.2025)

57. Роздільна здатність дисплея. Вікіпедія. стабільна версія перевірена 30 січня 2025 р. URL: https://uk.wikipedia.org/wiki/Роздільна_здатність_дисплея (дата звернення: 16.05.2025)

58. Гладченко О. Особливості формування інформаційної культури майбутніх фахівців цифрового дизайну (освітньо-професійна програма “Технології цифрового дизайну”) / [О. Гладченко, Т. Ратушняк, А. Омельчук, В. Ніжегородцев, Я. Вишемірська, О. Беспарточна] // Теорія і практика

управління соціальними системами, 2023. - № 4, С. 85-103.
DOI: <https://doi.org/10.20998/2078-7782.2023.4.08>

59. Програмування мовою JAVA: практикум: навчальний посібник / Ратушняк Т. В. - Державна фіскальна служба України, Університет державної фіскальної служби України. – Ірпінь, 2017. – 212 с.

60. Ратушняк Т.В. Інклюзивний вебдизайн: стандарти та цифрові інструменти тестування доступності вебсайту / [Т.В. Ратушняк, А.А. Омельчук, О.В. Гладченко, А.В. Хом'як, С.М. Хоцкіна] // Прикладні питання математичного моделювання, 2024. - № 2, Том 7. - С. 189-206. DOI: <https://doi.org/10.32782/mathematical-modelling/2024-7-2-17>

ДОДАТКИ

Додаток А (HTML файл)

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>TurboVive</title>
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Itim&display=swap" rel="stylesheet">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Ubuntu&display=swap" rel="stylesheet">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=PT+Sans&family=Ubuntu&display=swap"
rel="stylesheet">
  <link rel="stylesheet" href="main.css">
  <link rel="stylesheet" href="adaptive.css">
  <script src="script.js"></script>
</head>
<body>
<div class="main__container">
  <div class="header">
    <div id="mySidepanel" class="sidepanel">
      <a href="javascript:void(0)" class="closebtn" id="closeBtn">×</a>
      <a href="#" name="sydepanelNav">Головна</a>
      <a href="#service" name="sydepanelNav">Наші цінності</a>
      <a href="#gallery" name="sydepanelNav">Відповіді на часті запитання</a>
      <a href="#about-me" name="sydepanelNav">Чому ми</a>
      <a href="#contacts" name="sydepanelNav">Контакти</a>
    </div>
    <button class="openbtn" id="burgerBtn">☰</button>
    <div class="nav">
      <a href="#" class="nav__items">Головна</a>
      <a href="#service" class="nav__items">Наші цінності</a>
      <a href="#gallery" class="nav__items">Відповіді на часті запитання</a>
      <a href="#about-me" class="nav__items">Чому ми</a>
      <a href="#contacts" class="nav__items">Контакти</a>
    </div>

```

Продовження додатка А (HTML файл)

```

<div class="header__main js-scroll fade-in">
<div class="header__img js-scroll fade-in"></div>

<div class="header__text">
<div class="header__text__icons js-scroll ">
    <h2 class="name__item js-scroll slide-left " ></h2>
</div>
<p class="js-scroll fade-in" >
    <strong>TurboVive — надійний партнер</strong> у сфері відновлення
    турбін. Ми поєднуємо інженерну точність, сучасні технології
    та досвід, щоб повертати потужність кожному двигуну.
</p>
<span class=" js-scroll slide-left">Розпочнемо співпрацю?</span>
<div class="header__social__contacts__icons js-scroll fade-in">
    <a href="https://www.instagram.com/"></a>
    <a href="#contacts"></a>
    <a href="#contacts"></a>
</div>
</div>
</div>
</div>
<div class="service__block" id="service" >
    <h2 class="section-title">Наші цінності</h2>
    <div class="js-scroll fade-in">
        <h3 >Чесність</h3>
        
        <ul class="js-scroll fade-in-bottom">
            У TurboVive чесність є основою взаємин із клієнтами та партнерами.
            Ми відкрито інформуємо про стан обладнання, обсяги необхідного ремонту та
            вартість послуг.
            Наша мета — заслужити довіру через прозорість кожного процесу.
        </ul>
    </div>
    <div class="js-scroll fade-in">
        <h3 >Технічна експертиза</h3>
        
        <ul class="js-scroll fade-in-bottom">
            Команда TurboVive складається з досвідчених інженерів і технічних фахівців,
            які глибоко розуміють усі аспекти роботи турбокомпресорів.
            Ми не просто ремонтуємо — ми діагностуємо причини зносу,
            пропонуємо оптимальні технічні рішення і впроваджуємо їх за найвищими
            стандартами.
        </ul>
    </div>

```

Продовження додатка А (HTML файл)

```

        </ul>
    </div>
    <div class="js-scroll fade-in">
        <h3>Професійна гордість</h3>
        
        <ul class="js-scroll fade-in-bottom">
            Кожна відремонтована турбіна для нас — це приклад високої майстерності.
            TurboVive пишається результатами своєї роботи та прагне постійного
вдосконалення,
            впроваджуючи новітні технології та методики ремонту.
        </ul>
    </div>
</div>
<div class="gallery__block" id="gallery">
    <div class="gallery__text">
        <h2>Відповіді на часті запитання</h2>
    </div>
    <div class="articles">
        <div class="img01 js-scroll fade-in">
            
        </div>
        <div >
            <h3>Скільки часу триває ремонт турбіни?</h3>
            <p class="js-scroll slide-left">
                Тривалість ремонту турбіни залежить від типу поломки,
                ступеня зношення компонентів та наявності необхідних запчастин.
                У середньому стандартний ремонт займає від 1 до 3 робочих днів.
                У випадку складних пошкоджень або потреби у замовленні рідкісних деталей,
                строки можуть бути збільшені до 5–7 днів. Після діагностики ми завжди
                погоджуємо точний термін із клієнтом. Наша мета — відновити вашу турбіну
                швидко, якісно і без зайвих затримок.
            </p>
        </div>
    </div>
    <div>
        <h3>Чи надаєте ви гарантію на виконані роботи?</h3>
        <p>Так, ми впевнені у якості наших робіт, тому надаємо
        офіційну гарантію на ремонт турбін. Термін гарантії
        складає до 12 місяців або обмеження за пробігом
        (зазвичай до 20 000 км, залежно від умов експлуатації).
        Гарантія поширюється на відремонтовані або замінені
        компоненти та підтверджується відповідним документом.
        Ми завжди докладаємо максимум зусиль, щоб після
        реставрації ваша турбіна працювала надійно та довго.
    </p>
    </div>

```

Продовження додатка А (HTML файл)

```

</p>
</div>
  <div class="img02 js-scroll fade-in">
    
  </div>
  <div class="img01 js-scroll fade-in">
    
  </div>
  <div class="articles__text">
    <h3>Чи потрібно попередньо записуватися на діагностику?</h3>
    <p class="js-scroll slide-left">
      Так, попередній запис є бажаним. Це дозволяє нам оперативно
      планувати роботу майстрів та забезпечити швидке обслуговування
      без черг. Ви можете залишити заявку на сайті, зателефонувати
      або написати нам у месенджер. Якщо ситуація термінова, ми
      намагаємося піти назустріч і провести діагностику в день
      звернення. Попередня консультація допоможе уникнути
      простою вашого транспорту і заощадити ваш час.
    </p>
  </div>
  <div >
    <h3>Чи можна відремонтувати турбіну, якщо є тріщини в корпусі?</h3>
    <p class="js-scroll slide-left">
      Можливість ремонту тріщин корпусу турбіни залежить від
      характеру та місця пошкодження. Невеликі тріщини у
      вторинних зонах часто можна усунути методом аргонного
      зварювання або спеціальними технологіями відновлення.
      Якщо ж ушкодження зачіпає критичні елементи або впливає
      на баланс турбіни, ми рекомендуємо заміну відповідної
      частини. Наші фахівці проводять детальний аналіз і
      пропонують оптимальне рішення: ремонт або підбір
      іюригінальних або аналогових комплектуючих.
    </p>
  </div>
  <div class="img01 js-scroll fade-in">
    
  </div>
</div>
<div id="opened" class="opened">
  <div class="opened-img">
    <div class="closed">&times;</div>
    <img src="" alt="">
  </div>
</div>

```

Продовження додатка А (HTML файл)

```

</div>
<div class="about__me__block" id="about-me">
  <div class="about__me__text js-scroll fade-in-bottom">
    <h3>Чому ми</h3>
    <div class="about__me__content">
      <div class="about__me__copy">
        <p><strong>TurboVive — експерти у відновленні потужності вашого
двигуна</strong></p>
        <p>Команда TurboVive поєднує глибокі технічні знання з багаторічним практичним
досвідом, щоб для кожної турбіни знайти оптимальне рішення реставрації та ремонту.
        Ми відновлюємо турбіни з урахуванням індивідуальних особливостей їх роботи,
        застосовуючи перевірені технології та власні напрацювання у сфері обслуговування
автомобільної техніки.</p>
        <p>З 2015 року ми працюємо як сертифікований сервісний центр, що спеціалізується
на ремонті турбокомпресорів
        усіх типів. Наші майстри мають за плечима тисячі успішно відновлених турбін,
впровадження авторських
        методик ремонту та участь у розробці програм підвищення ефективності агрегатів.
Ми віддані <strong>якості,
        точності й надійності </strong> у кожному ремонті.</p>
      </div>
      <div class="about__me__img">
        
      </div>
    </div>
  </div>
</div>
<div class="free__space__block" >
</div>
<div class="contacts__block" id="contacts">
  <div class="contacts__overlay">
    <div class="contacts__form">
      <h3>Контакти</h3>
      <p><strong>Адреса:</strong>вул. Університетська, 51, м. Ірпінь</p>
      <p><strong>E-mail:</strong>turbovive@gmail.com</p>
      <p><strong>Номер телефону:</strong>+380 63 373 62 38</p>
      <h4>Графік роботи</h4>
      <p>Пн–Сб: 10:00 – 20:00<br>Нд: 11:00 – 19:00</p>
    </div>
  </div>
</div>
<footer>
  <p>© 2025 «TurboVive». Всі права захищено</p>
</footer>
</div>
</body>

```

Додаток Б (CSS файл, базові стилі)

```

</html>
html{
    scroll-behavior: smooth;
}
.main__container{
    width: 100%;
    display: grid;
    align-items: start;
    grid-template-rows: repeat(auto-fill, 1fr);
    grid-template-columns: repeat(3, 1fr);
    overflow-x: hidden;
    overflow-y: hidden;
    font-family: 'PT Sans', sans-serif;
    font-family: 'Ubuntu', sans-serif;
}
.header {
    background: url('./Img__background/turbo.png') center no-repeat;
    background-attachment: fixed;
    background-size: cover;
    width: 100%;
    height: 800px;
    margin: auto;
    grid-column: 1/4;
    grid-row: 1;
}
.nav{
    width: 100%;
    height: 5%;
    background-color: rgb(101, 119, 163);
    position: fixed;
    top: 0;
    left: 0;
    z-index: 4;
    display: flex;
    justify-content: center;
    align-items: center;
}
.nav__items{
    color: whitesmoke;
    margin-left: 58px;
    text-transform: uppercase;
    font-weight: 600;
    font-size: 15px;
}
.nav__items:hover{
    color: rgb(190, 185, 185);
    animation: 2s show ease;
}
#service {
    scroll-margin-top: 60px;
}

```

Продовження додатку Б (CSS файл, базові стилі)

```

}
#gallery{
  scroll-margin-top: 40px;
}
@keyframes show {
  from { opacity: 0.4; }
  to { opacity: 1; }
}
.header__main{
  width: 100%;
  height: 500px;
  background: linear-gradient(90deg, rgba(127, 125, 128, 0.6) 0%, rgba(255,255,255,0.6) 100%);
  display: flex;
  flex-direction: row;
  margin-top: 100px;
  margin-left: 140px;
}
.header__text{
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  align-content: space-around;
  width: 50%;
  margin-top: 217px;
  margin-left: -20px;
  z-index: 3;
  height: 100px;
}
.header__img{
  width: 429px;
  height: 456px;
  /* background-color: rgba(157, 146, 164, 0.576); */
  border-radius: 2px;
  opacity: 0;
  margin-left: 115px;
  margin-top: 102px;
}
.header__img img {
  width: 101%;
  height: 63%;
  border-radius: 2.5px;
  opacity: 1;
}
.header__text img{
  opacity: 0.6;
}
.header__text h2{
  margin-top: 10px;
  text-transform: uppercase;
}

```

Продовження додатку Б (CSS файл, базові стилі)

```

/* font-size: 18px; */
}
.header__text p{
    text-align: justify;
    width: 500px;
    height: 133px;
    color: rgb(0, 0, 0);
    font-size: 21px;
    margin-left: 70px;
    margin-top: 50px;
}
.header__text span{
    color: rgb(0, 0, 0);
    font-size: 21px;
    margin-top: 50px;
    margin-bottom: 200px;
    margin-left: 70px;
}
.header__text__icons{
    box-sizing: border-box;
}
.header__text__icons img{
    margin-top: 10px;
}
.header__social__contacts__icons{
    display: flex;
    justify-items: center;
    margin-left: 4%;
    margin-top: -13%;
}
.header__welcome__item{
    color: rgb(0, 0, 0);
    margin-left: 58px;
    /* margin-top: 33px; */
}
.name__item{
    color: white;
    margin-left: 20px;
}
.header__text__icon h2{
    margin-left: 100px;
}
.header__social__contacts__icons img{
    width: 46px;
    height: 46px;
    border-radius: 50%;
    margin-left: 15px;
    opacity: 0.7;
    background-color: rgba(63 107 216 / 41%);
}

```

Продовження додатку Б (CSS файл, базові стилі)

```
.header__social__contacts__icons img:hover{
  opacity: 0.7;
  background-color: whitesmoke;
}
a{
  text-decoration: none;
}
.service__block{
  display: flex;
  position: relative;
  background-color: #fff;
  /*background: url('./gallery/5.jpg') no-repeat center;*/
  opacity: 0.9;
  background-attachment: fixed;
  background-size: cover;
  width: 100%;
  height: 800px;
  grid-column: 1/4;
  grid-row: 2;
  justify-content: center;
  align-items: flex-start;
  margin-top: 29px;
  flex-wrap: wrap;
  justify-content: center;
  gap: 27px;
}
.service__block div{
  width: 300px;
  height: 512px;
  margin-top: -165px;
  border-radius: 2%;
  opacity: 0.9;
  padding-left: 5px;
  background-color: rgba(75, 125, 218, 0.674);
  height: 50%;
}
.section-title {
  flex: 0 0 100%;
  width: auto;
  text-align: center;
  text-transform: uppercase;
  font-size: 32px;
  font-weight: 700;
  margin: 60px 0 40px;
  margin-bottom: -90px;
  margin-top: 35px;
}
.first__icon{
  width: 75px;
  height: 56px;
```

Продовження додатку Б (CSS файл, базові стилі)

```

    position: relative;
    display: block;
    margin: 0 auto;
    margin-top: 5%;
    margin-bottom: 7%;
}
.first__icon1 {
    width: 60px;
    height: 60px;
    position: relative;
    display: block;
    margin: 0 auto;
    margin-top: 5%;
    margin-bottom: 7%;
}
.first__icon2 {
    width: 60px;
    height: 60px;
    position: relative;
    display: block;
    margin: 0 auto;
    margin-top: 5%;
    margin-bottom: 7%;
}
.service__block div>img{
    opacity: 0.9;
}
.service__block div ul{
    margin-top: -10px;
}
.service__block ul{
    color: rgb(0, 0, 0);
    font-size: 17px;
    font-weight: 500;
    padding-right: 10px;
    text-align: justify;
    margin-left: -34px;
}
.service__block h3 {
    color: rgb(0, 0, 0);
    width: 101%;
    border-bottom: 3px solid rgb(15, 14, 14);
    padding-top: 15px;
    text-align: center;
    display: block;
    margin: 0 auto;
    padding-bottom: 15px;
    margin-left: -4px;
}
.service__block h2{

```

Продовження додатку Б (CSS файл, базові стилі)

```

    color:rgb(0, 0, 0);
}
.service__block span{
    width: auto;
    color: white;
    position: absolute;
    top: 600px;
    right: 560px;
    font-size: 17px;
}
.service__block span>p{
    margin-left: 30px;
    margin-bottom: -10px;
}
.articles {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(320px, 1fr));
    grid-auto-rows: auto;
    gap: 40px 50px;
    margin: 0 auto 80px;
    max-width: 1000px;
    color: #555;
    font-family: 'Ubuntu', sans-serif;
}
.articles img {
    width: 100%;
    height: auto;
    border-radius: 6px;
    box-shadow: 0 4px 12px rgba(0,0,0,0.1);
}
.articles h3 {
    margin-bottom: 8px;
    font-size: 18px;
    color: #333;
}
.articles p {
    line-height: 1.6;
    text-align: justify;
    font-weight: 700;
}
.gallery__text {
    text-align: center;
    color: #000000;
    text-transform: uppercase;
    font-size: 20px;
    margin: 60px auto 40px;
    max-width: 90%;
}
.gallery__block {
    grid-column: 1 / 4;
}

```

Продовження додатку Б (CSS файл, базові стилі)

```
.about__me__block {
  grid-column: 1 / -1;
  background: url('/gallery/turbo2.png') center/cover no-repeat;
  padding: 80px 20px;
  position: relative;
  overflow: hidden;
}
.about__me__text {
  max-width: 1200px;
  margin: 0 auto;
  background: rgba(255,255,255,0.85);
  border-radius: 8px;
  box-shadow: 0 8px 24px rgba(0,0,0,0.15);
  padding: 40px;
}
.about__me__text h3 {
  text-align: center;
  font-size: 28px;
  text-transform: uppercase;
  margin-bottom: 30px;
  color: #111;
}
.about__me__content {
  display: flex;
  align-items: center;
  gap: 5%;
  flex-wrap: unset;
}
.about__me__copy {
  flex: 1 1 400px;
  color: #222;
  line-height: 1.6;
  text-align: justify;
  font-weight: 600;
}
.about__me__copy p {
  margin-bottom: 16px;
  font-size: 16px;
}
.about__me__img {
  flex: 0 0 300px;
  text-align: center;
}
.about__me__img img {
  max-width: 100%;
  border-radius: 6px;
  box-shadow: 0 4px 16px rgba(0,0,0,0.1);
  height: auto;
}
.free__space__block{
  background: linear-gradient(90deg, rgba(127, 125, 128, 0.6) 0%, rgba(255, 255, 255, 0.6) 100%);
```

Продовження додатку Б (CSS файл, базові стилі)

```

background-attachment: fixed;
background-size: cover;
width: 100%;
height: 100px;
grid-column: 1/4;
grid-row: 5;
opacity: 0.9;
}
.contacts__block {
  position: relative;
background: url('../gallery/turbo3.png') center/cover no-repeat;
min-height: 100vh;
display: flex;
justify-content: center;
align-items: center;
padding: 40px 20px;
  grid-column: 1 / 4;
}
.contacts__block::before {
  content: "";
  position: absolute;
  inset: 0;
background: linear-gradient(135deg, rgba(0,0,0,0.4), rgba(0,0,0,0.1));
  z-index: 1;
}
.contacts__overlay {
  position: relative;
  z-index: 2;
  max-width: 450px;
  width: 100%;
background: #fff;
border-radius: 12px;
padding: 40px 30px;
box-shadow: 0 12px 40px rgba(0,0,0,0.2);
  text-align: center;
}
.contacts__form h3 {
  font-size: 28px;
margin-bottom: 24px;
color: #333;
text-transform: uppercase;
  letter-spacing: 1px;
}
.contacts__form p {
  margin: 12px 0;
  line-height: 1.6;
font-weight: 600;
color: #535252;
}
.contacts__form p strong {
  display: block;

```

Продовження додатку Б (CSS файл, базові стилі)

```

    margin-bottom: 6px;
    font-weight: 600;
    color: #222;
  }
  footer {
    width: 100%;
    height: 70px;
    background-color: rgb(56, 55, 55);
    grid-column: 1/4;
    grid-row: 7;
  }
  footer div {
    display: flex;
    justify-content: center;
    align-content: center;
    margin: auto;
  }
  footer img {
    border-radius: 50%;
    background-color: white;
    /* opacity: 0.9; */
    width: 45px;
    height: 45px;
    margin: 20px 7px;
  }
  footer p {
    color: white;
    text-align: center;
    margin-top: 25px;
  }
  .openbtn {
    display: none;
    position: fixed;
    top: 20px;
    left: 20px;
    z-index: 999;
    background-color: #4a6ea9;
    color: white;
    padding: 10px 15px;
    font-size: 26px;
    border: none;
    border-radius: 4px;
    cursor: pointer;
  }
  .sidepanel {
    position: fixed;
    top: 0;
    left: 0;
    height: 100vh;
    width: 0; background-color: rgba(0, 0, 0, 0.9);

```

Продовження додатку Б (CSS файл, базові стилі)

```

overflow-x: hidden;
transition: 0.3s;
z-index: 9999;
display: flex;
flex-direction: column;
padding-top: 60px;
}
.sidepanel a {
padding: 12px 24px;
text-decoration: none;
font-size: 22px;
color: white;
display: block;
transition: 0.2s;
}
.sidepanel a:hover {
background-color: rgba(255, 255, 255, 0.1);
}
.sidepanel .closebtn {
position: absolute;
top: 20px;
right: 25px;
font-size: 36px;
color: white;
cursor: pointer;
}
.js-scroll {
opacity: 0;
transition: 500ms;
}
.js-scroll.scrolled {
opacity: 1;
}
.scrolled.fade-in {
animation: 2.5s show ease-in-out both;
}
.scrolled.fade-in-bottom {
animation: 1s hide ease-in-out both;
}
.scrolled.slide-left {
animation: 1.5s move ease-in-out both;
}
.scrolled.slide-right {
animation: 1.5s move ease-in-out both;
}
@keyframes move {
0% {
-webkit-transform: translateX(-100px);
transform: translateX(-100px);
opacity: 0;
}
}

```

Продовження додатку Б (CSS файл, базові стилі)

```
100% {  
  -webkit-transform: translateX(0);  
  transform: translateX(0);  
  opacity: 1;  
}  
}  
@keyframes move {  
  0% {  
    -webkit-transform: translateX(100px);  
    transform: translateX(100px);  
    opacity: 0;  
  }  
  100% {  
    -webkit-transform: translateX(0);  
    transform: translateX(0);  
    opacity: 1;  
  }  
}  
@keyframes hide {  
  0% {  
    -webkit-transform: translateY(50px);  
    transform: translateY(50px);  
    opacity: 0;  
  }  
  100% {  
    -webkit-transform: translateY(0);  
    transform: translateY(0);  
    opacity: 1;  
  }  
}  
@keyframes show {  
  0% {  
    opacity: 0;  
  }  
  100% {  
    opacity: 1;  
  }  
}
```

Додаток Г(Java Script)

```

window.onload = function() {
  let openedWindow = document.getElementById("opened");
  let openedImg = document.querySelector(".opened-img img");
  let closeButton = document.getElementsByClassName("closed")[0];
  let images = document.getElementsByName('image');
  for (let i = 0; i < images.length; i++) {
    images[i].addEventListener("click", imageClick, false);
  }
  closeButton.addEventListener("click", function () {
    openedWindow.style.display = "none";
  })
  function imageClick(e) {
    let img = e.target;
    openedWindow.style.display = "block";
    openedImg.src = img.currentSrc;
    openedImg.style.width = img.naturalWidth;
    openedImg.style.height = img.naturalHeight;
  }
  const scrollElements = document.querySelectorAll(".js-scroll");
  const elementInView = (el, dividend = 1) => {
    const elementTop = el.getBoundingClientRect().top;
    return (
      elementTop <=
      (window.innerHeight || document.documentElement.clientHeight) / dividend
    );
  };
  const elementOutOfView = (el) => {
    const elementTop = el.getBoundingClientRect().top;
    return (
      elementTop > (window.innerHeight || document.documentElement.clientHeight)
    );
  };
  const displayScrollElement = (element) => {
    element.classList.add("scrolled");
  };
  const hideScrollElement = (element) => {
    element.classList.remove("scrolled");
  };
  const handleScrollAnimation = () => {
    scrollElements.forEach((el) => {
      if (elementInView(el, 1.25)) {
        displayScrollElement(el);
      } else if (elementOutOfView(el)) {
        hideScrollElement(el);
      }
    })
  }
  window.addEventListener("scroll", () => {
    handleScrollAnimation();
  });
  handleScrollAnimation()
  let burgerBtn = document.getElementById('burgerBtn');
  burgerBtn.addEventListener('click', openNav);
  function openNav() {
    document.getElementById("mySidepanel").style.width = "100%";
  }
}

```

Продовження додатку Г (Java Script)

```
document.getElementById("mySidepanel").style.height = "100%";
burgerBtn.style.visibility = 'hidden';
}
let closeBtn = document.getElementById('closeBtn');
closeBtn.addEventListener('click', closeNav, false);
function closeNav() {
    document.getElementById("mySidepanel").style.width = "0%";
    document.getElementById('burgerBtn').style.visibility = 'visible';
}
let sydepanelNav = document.getElementsByName('sydepanelNav');
for(let i = 0; i<sydepanelNav.length; i++){
    sydepanelNav[i].addEventListener("click", closeNav, false);
}
}
```

Додаток В (Мобільні телефони ($\leq 480\text{px}$))

```

/* Мобільні телефони (до 480px) */
@media (max-width: 480px) {
  .header__main {
    flex-direction: column;
    margin-left: 0;
    padding: 1px;
    height: 698px;
  }
  .header__img {
    margin: 0 auto;
    width: 280px;
    height: auto;
  }
  .header__img img {
    width: 100%;
    height: auto;
    position: relative;
    margin-top: 17%;
  }
  .header__text {
    width: 79%;
    margin: 20px auto;
    padding: 0px 0px;
    height: 44%;
    text-align: justify;
    display: flex;
    align-items: center;
    justify-content: center;
  }
  .header__text p,
  .header__text span {
    width: 100%;
    margin-left: 0;
    text-align: center;
  }
  .header__text p {
    font-size: 18px;
    height: auto;
  }
  .header__welcome__item {
    font-size: 22px;
    margin-left: 0;
    text-align: center;
  }
  .header__social__contacts__icons {
    justify-content: center;
    margin: 30px auto;
    flex-wrap: wrap;
    margin-top: -44%;
  }
  .section-title {
    flex: 0 0 100%;
    width: auto;
    text-align: center;
  }

```

Продовження додатку В (Мобільні телефони ($\leq 480px$))

```

    text-transform: uppercase;
    font-size: 32px;
    font-weight: 700;
    margin: 60px 0 40px;
    margin-bottom: 20px;
    margin-top: 12px;
}
.service__block {
    height: auto;
    padding: 0px;
    display: flex;
    justify-content: center;
}
.service__block h2 {
    width: 100%;
    height: 100%;
    display: flex;
    align-items: center;
    align-content: center;
    justify-content: center;
    overflow: auto;
    /*text-align: center;
    color: #000000;
    text-transform: uppercase;
    font-size: 20px;
    margin: 60px auto 40px;
    max-width: 90%; */
}
.service__block div {
    margin-top: 0;
    width: 90%;
    height: auto;
}
.articles {
    grid-template-columns: 1fr;
    padding: 0 10px;
}
.about__me__content {
    flex-direction: column;
}
.contacts__overlay {
    padding: 20px;
}
}

```

Додаток Г (Планишети (481-768px))

```

/* Планишети (481px - 768px) */
@media (min-width: 481px) and (max-width: 768px) {
  .nav {
    display: none;
  }
  .header__main {
    flex-direction: column;
    margin-left: 0;
    padding: 0px;
    height: 700px;
  }
  .header__img {
    margin: 0 auto;
    width: 320px;
    height: auto;
  }
  .header__img img {
    width: 101%;
    height: auto;
    border-radius: 2.5px;
    opacity: 1;
  }
  .header__text {
    margin: 20px auto;
    width: 90%;
    height: 50%;
    padding: 0;
  }
  .header__text p,
  .header__text span {
    width: 100%;
    margin-left: 0;
    text-align: center;
  }
  .header__social__contacts__icons {
    margin: 30px auto 0;
    justify-content: center;
    margin-top: -9%;
  }
  .service__block {
    height: auto;
    padding: 0px;
    display: flex;
    justify-content: center;
    margin-top: 6%;
  }
  .service__block div {
    margin-top: 0;
    width: 91%;
    height: auto;
  }
  .about__me__content {
    flex-direction: column;
  }
}

```

Продовження додатку Г (Планишети (481-768px))

```
.section-title {  
  flex: 0 0 100%;  
  width: 100%;  
  text-align: center;  
  text-transform: uppercase;  
  font-size: 32px;  
  font-weight: 700;  
  margin: 20px auto 40px;  
}  
}
```

Додаток Д (Ноутбуки та середні екрани (769px - 1199px))

```

/* Ноутбуки та середні екрани (769px - 1199px) */
@media (min-width: 769px) and (max-width: 1199px) {
  .nav__items {
    color: whitesmoke;
    margin-left: 3%;
    text-transform: uppercase;
    font-weight: 600;
    font-size: 100%;
  }
  .header__main {
    margin-left: 20px;
    height: 34em;
  }
  .header__text p {
    text-align: justify;
    width: 71%;
    height: 133px;
    color: rgb(0, 0, 0);
    font-size: 21px;
    margin-left: -118px;
    margin-top: -84px;
  }
  .header__text span {
    color: rgb(0, 0, 0);
    font-size: 21px;
    margin-top: 24%;
    margin-bottom: 56%;
    margin-left: -113px;
  }
  .header__social__contacts__icons{
    display: flex;
    justify-items: center;
    margin-left: -8em;
    margin-top: -46%;
  }
  .header__img {
    width: 350px;
    height: auto;
  }
  .header__img img {
    width: 100%;
    height: auto;
    margin-left: -77px;
  }
  .header__text {
    width: 50%;
    margin-top: 26%;
    margin-left: 1em;
  }
  .service__block {
    display: flex;
    justify-content: center;
    align-items: flex-start;
    gap: 202px;
  }

```

Продовження додатку Д (Ноутбуки та середні екрани (769px – 1199px))

```
padding: 0px;
flex-wrap: wrap;
margin-bottom: -15%;
}
.section-title {
margin-top: 20px;
padding-bottom: 96px;
}
.service__block div {
flex: 0 0 300px;
width: 32%;
height: 50%;
}
.gallery__block {
grid-column: 1 / 4;
margin-top: 670px;
}
}
```

Додаток Е (Великі екрани (від 1200px))

```
/* Великі екрани (1200px+) */  
@media (min-width: 1200px) {  
  .main__container {  
    max-width: 100%;  
    margin: 0 auto;  
  }  
  .header__main {  
    height: 500px;  
  }  
}  
.openbtn {  
  display: none;  
  position: fixed;  
  top: 20px;  
  left: 20px;  
  z-index: 1000;  
  background-color: #4a6ea9;  
  color: white;  
  padding: 10px 15px;  
  font-size: 26px;  
  border: none;  
  border-radius: 4px;  
  cursor: pointer;  
}
```

Додаток Є (Результати аудиту Google Lighthouse)

