

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики

(інститут)

Факультет інформаційних технологій

(факультет)

Кафедра Програмного забезпечення комп'ютерних систем

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента

Жовнір Карини Олександрівни

(ПІБ)

академічної групи

121-19-1

(шифр)

спеціальності

121 Інженерія програмного забезпечення

(код і назва спеціальності)

освітньої програми

Інженерія програмного забезпечення

(назва освітньої програми)

на тему:

Розробка веб-сайту для людей з обмеженими

можливостями

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	<i>проф. Лактіонов І.С.</i>			
розділів:				
спеціальний	<i>проф. Лактіонов І.С.</i>			
економічний	<i>доц. Касьяненко Л.В.</i>			
Рецензент				
Нормоконтролер	<i>ст. викл. Мартиненко А.А.</i>			

Дніпро
2023

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:

завідувач кафедри

програминого забезпечення комп'ютерних систем

(повна назва)

М.О. Алексєєв

(підпис)

(прізвище, ініціали)

« » 2023 року

ЗАВДАННЯ

на кваліфікаційну роботу

бакалавра

(назва освітньо-кваліфікаційного рівня)

студенки 121-19-1

(група)

Жовнір Карини Олександрівни

(прізвище та ініціали)

тема кваліфікаційної роботи

Розробка веб-сайту для людей з обмеженими

можливостями

затверджена наказом ректора НТУ «ДП» від

16.05.2023

№ 350-с

Розділ	Зміст виконання	Термін виконання
Спеціальний	<i>На основі матеріалів проєктно-технологічної та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми</i>	<i>13.05.2023 р.</i>
Економічний	<i>Провести розрахунок трудомісткості розробки програминого забезпечення, витрат на створення ПЗ й тривалості його розробки</i>	<i>27.05.2023 р.</i>

Завдання видав

проф.Лактіонов І. С.

(підпис)

(посада, прізвище, ініціали)

Завдання прийняв до виконання

Жовнір К.О.

(підпис)

(прізвище, ініціали)

Дата видачі завдання: 14.01.2023 р.

Термін подання кваліфікаційної роботи до ЕК: 12.06.2023 р.

РЕФЕРАТ

Пояснювальна записка: 99 с., 48 рис., 3 табл., 3 дод., 26 джерел.

Об'єкт розробки: веб-сайт з урахуванням можливостей доступу для людей з різними фінансовими, фізичними та когнітивними обмеженнями.

Мета кваліфікаційної роботи: розробка сайту для людей з обмеженими можливостями, а також розширення та інклюзії контенту в Інтернеті за допомогою різних доступних практик.

У вступі була висвітлено соціальну проблему інклюзії в Інтернеті, підкреслена важливість цієї теми в сучасному світі, зокрема в контексті політичних та соціальних реалій. Це створило фундаментальну базу для подальшого дослідження та розробки проекту.

У першому розділі проведено аналіз українських веб-сайтів, зокрема тих, які мають значний потік користувачів. Цей аналіз допоміг виявити помилки та проблеми доступності на конкуруючих сайтах, що в свою чергу допомогло уникнути аналогічних проблем у розробці власного проекту.

У другому розділі було обрано необхідні технічні засоби та програми для розробки проекту. Було надано схеми, опис роботи сайту, а також детальний опис функціональності. Особливу увагу було приділено практикам доступності, що дозволило забезпечити широкий доступ до сайту для різних верств населення.

У економічному розділі проведено розрахунок трудомісткості розробки програмного забезпечення, вартості робіт, а також визначено час, необхідний для створення проекту. Це дало можливість оцінити економічний аспект проекту та планувати фінансові ресурси.

Практичне значення кваліфікаційної роботи полягає в створенні комерційного сайту з контактною формою, який забезпечить доступність та включення користувачів з різними обмеженнями. Це відкриває можливості для розширення та поліпшення доступу до веб-контенту для всіх користувачів.

Актуальність кваліфікаційної роботи визначається морально-соціальними принципами та правами кожного користувача Інтернету на доступ до інформації та веб-ресурсів. Розробка сайту з урахуванням доступності є важливим кроком у забезпеченні рівних можливостей та інклюзивного середовища в Інтернеті.

Ключові слова: ІНТЕРНЕТ, ВЕБ-САЙТ, ІНКЛЮЗИЯ, ДОСТУПНІСТЬ, РОЗРОБКА САЙТІВ, ДОСТУПНІСТЬ В МЕРЕЖІ, ОБМЕЖЕНІ МОЖЛИВОСТІ, ІНКЛЮЗИВНЕ СЕРЕДОВИЩЕ, ПРАКТИКИ ПО ДОСТУПНОСТІ.

ABSTRACT

Explanatory note: 99 pages., 48 img., 3 table, 3 appendix, 26 sources.

Development object: A website with considerations for accessibility for individuals with diverse financial, physical, and cognitive limitations.

Qualification work objective: Developing a website for people with disabilities, as well as expanding and inclusive content on the internet through various available practices.

The introduction highlighted the social problem of inclusivity on the internet, emphasizing the importance of this topic in the modern world, particularly in the context of political and social realities. This created a fundamental basis for further research and project development.

The first chapter analyzed Ukrainian websites, particularly those with a significant user traffic. This analysis helped identify accessibility errors and issues on competing websites, thereby avoiding similar problems in the development of the project.

The second chapter selected the necessary technical tools and programs for project development. It provided diagrams, a website working description, and a detailed functionality description. Special attention was given to accessibility practices, ensuring broad access to the website for different segments of the population.

In the economic section, the calculation of software development effort, cost of work, and the time required to create the project were conducted. This allowed for an assessment of the project's economic aspect and the planning of financial resources.

The practical significance of the qualification work lies in the creation of a commercial website with a contact form, ensuring accessibility and inclusion of users with various limitations. This opens up possibilities for expanding and improving access to web content for all users.

The relevance of the qualification work is determined by the moral-social principles and rights of every Internet user to access information and web resources. Developing an accessible website is an important step in ensuring equal opportunities and an inclusive environment on the Internet.

Keywords: INTERNET, WEBSITE, INCLUSION, ACCESSIBILITY, WEBSITE DEVELOPMENT, WEB ACCESSIBILITY, DISABILITIES, INCLUSIVE ENVIRONMENT, ACCESSIBILITY PRACTICES.

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ЗМІСТ	5
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	Ошибка! Закладка не определена.
ВСТУП.....	Ошибка! Закладка не определена.
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ	
Ошибка! Закладка не определена.	
1.1. Загальні відомості з предметної галузі	Ошибка! Закладка не определена.
1.2 Призначення розробки та галузь застосування.....	24
1.3. Підстава для розробки	24
1.4. Постановка завдання.....	24
1.5. Вимоги до програми або програмного виробу.....	25
1.5.1. Вимоги до функціональних характеристик.....	25
1.5.2. Вимоги до інформаційної безпеки	25
1.5.3. Вимоги до складу та параметрів технічних засобів	26
1.5.4. Вимоги до інформаційної та програмної сумісності	Ошибка! Закладка не определена.
определена.	
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ	
Ошибка! Закладка не определена.	
2.1 Функціональне призначення програми	Ошибка! Закладка не определена.
2.2 Опис застосованих математичних методів.....	28
2.3 Опис використаної архітектури та шаблонів проектування.....	29
2.4 Опис використаних технологій та мов програмування.....	35

2.5	Опис структури програми та алгоритмів її функціонування.....	41
2.6	Обґрунтування та організація вхідних та вихідних даних програми	46
2.7	Опис розробленого програмного продукту.....	50
2.7.1	Використані технічні засоби.....	50
2.7.2	Використані програмні засоби.....	50
2.7.3	Виклик та завантаження програми.....	51
2.7.4	Опис інтерфейсу користувача.....	52
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ		67
3.1	Розрахунок трудомісткості та вартості розробки програмного продукту	67
3.2.	Розрахунок витрат на створення програми	71
ВИСНОВКИ.....		73
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ		74
Додаток А. Лістинг програми		76
Додаток Б. Відгук керівника економічного розділу		100
Додаток В. Перелік документів на оптичному носії		101

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ВП або WP – CMS WordPress;

CRM (Customer relationship management) – система управління взаємовідносинами з клієнтами;

БД – база даних;

W3C (The World Wide Web Consortium) – всесвітній веб-консорціум;

ARIA (Accessible Rich Internet Application) — доступний багатофункціональний інтернет-додаток.

ВСТУП

Завдання даної кваліфікаційної роботи та об'єкта її діяльності безпосередньо пов'язані з напрямом «Інформаційні технології» та відповідає узагальненій тематиці кваліфікаційних робіт і переліку зазначених виробничих функцій, типових задач діяльності, умінню та компетенціям, якими повинні володіти бакалаври спеціальності 121 «Інженерія програмного забезпечення».

Темою даної кваліфікаційної роботи є розробка веб-сайту з використанням допоміжних методів для людей з обмеженими можливостями.

Метою даної кваліфікаційної роботи є розробка веб-сайту для людей з обмеженими можливостями за рахунок використання теоретико-прикладних засад HTML-розмітки, каскадних таблиць стилів (CSS), мови програмування JavaScript. Використання стандартів Консорціум Всесвітньої павутини (W3C) по забезпеченню доступності контенту, що дозволяє використовувати сайт людям з обмеженими можливостями, при необхідності використовуючи клавіатуру для навігації по сайту або синтезатор мови.

В наш час дуже важливо створювати сайти для людей з обмеженими можливостями, щоб кожен мав змогу почувати себе повноцінним та мати доступ до будь-яких ресурсів глобальної мережі Інтернет. На сьогоднішній день переважна більшість веб-сайтів не відповідають стандартам по забезпеченню доступності. Ця проблема стає помітною навіть після аналізу офіційних веб-сайтів що призначені для надання різноманітних послуг громадянам України. На сьогодні в Україні немає єдиних стандартів, вимог або методичних рекомендацій щодо розробки веб-сайтів, адаптованих для людей з проблемами зору. Тому єдиним варіантом залишається використання норм та рекомендацій міжнародного рівня та вивчення закордонного досвіду із запуску веб-сайтів та сервісів. Тому важливо

вносити цю тему на розголос а також агітувати до застосування веб доступності на всіх сайтах.

Отже, результатом даної кваліфікаційної роботи є веб сайт, в якому використано стандарти Консорціум Всесвітньої павутини (W3C) по забезпеченню доступності контенту для людей з обмеженими можливостями.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Загальні відомості з предметної галузі

На даний час Інтернет широко використовують як джерело для пошуку інформації, навчання, дозвілля та спілкування. Всесвітню павутину, пошукові системи, веб сторінки та веб додатки масово використовуються людьми будь якого віку та статті, а також людьми з різними можливостями. Зручність, сприйняття контенту та час перебування на веб сторінці залежать не лише від написаного текст, а також від дизайну та доступності для різних вікових категорій, соціального статусу та загальних можливостей.

На даний час в Україні інклюзія в Інтернеті не розвинена на тому рівні, якого потребує наш час. Після повномасштабного вторгнення в Україну ця проблема стає більш гострішою, а тому слід підіймати її частіше.

Проаналізувавши популярні веб-сторінки, легко можна знайти сайти, які використовуються для соціального вжитку, але не є повністю оснащеними додатковими характеристиками, які б надавали повну доступність людям з різними обмеженнями.

Наприклад, компанія ТОВ «Ясно» [1], яка забезпечує електроенергією міста-мільйонники Київ та Дніпро мають проблеми з зчитування з екрану, а саме фокусування на прихованих посиланнях(рис. 1.1).

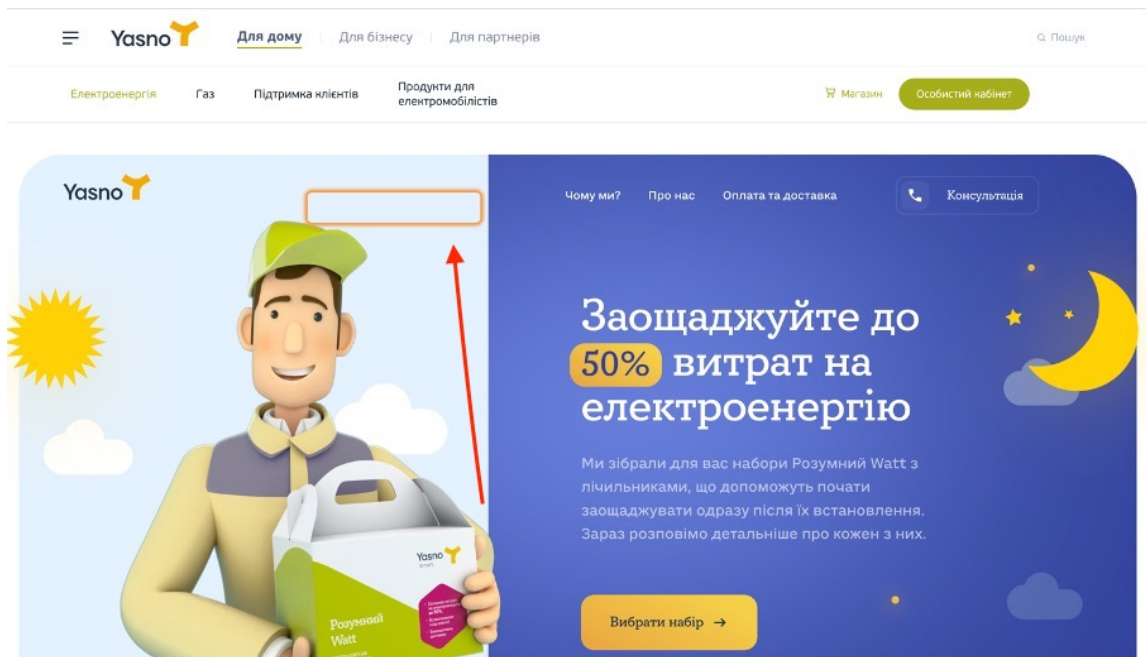


Рис. 1.1. Фокусування на прихованих посиланнях

Також «Ясно» недостатньо подбали про людей з поганим зором, так як на деяких ділянках сайту контрастність між текстом та фоном замала (рис. 1.2 та рис. 1.3).

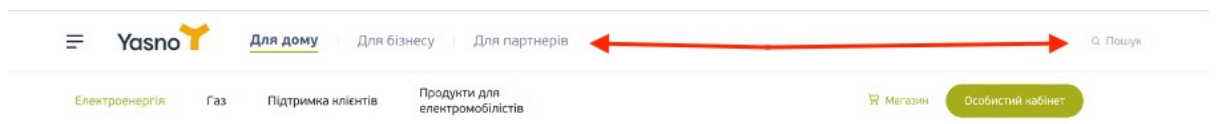


Рис. 1.2. Приклад №1 недостатньої контрастності між текстом та фоном

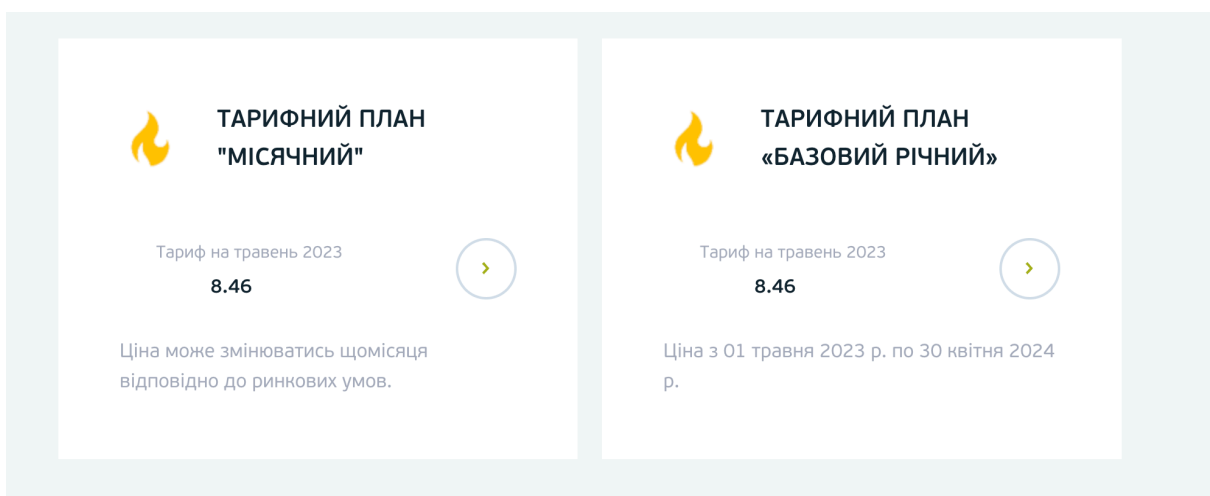
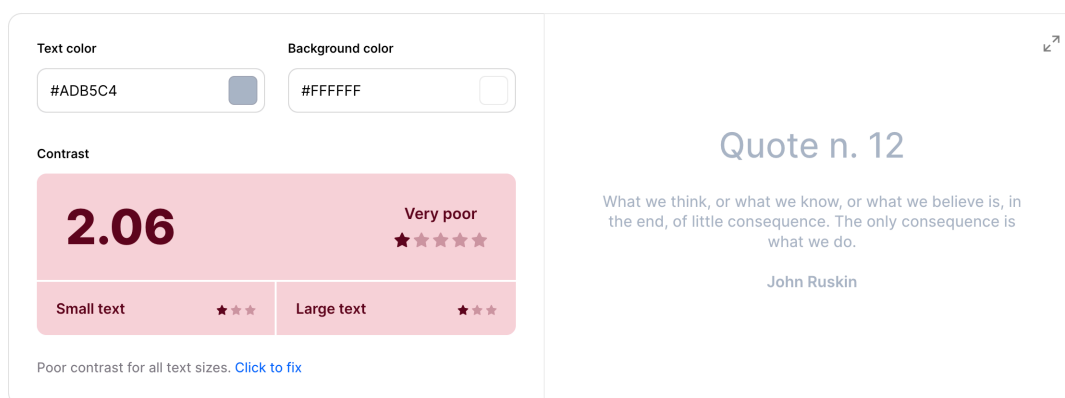


Рис. 1.3. Приклад №2 недостатньої контрастності між текстом та фоном

Перевірку було зроблено за допомогою веб-сервісу [2], який відповідає вимогам посібника із забезпечення доступності веб-контенту. Використовуючи відкритий код сайту, було отримано інформацію про колір тексту на сайті та перевірено контрастність. Отримані результати не є задовільними (рис. 1.4).

Color Contrast Checker

Calculate the contrast ratio of text and background colors.



The screenshot shows the Color Contrast Checker interface. On the left, there are input fields for 'Text color' (hex code #ADB5C4) and 'Background color' (hex code #FFFFFF). Below these, the 'Contrast' section displays a large red '2.06' and the text 'Very poor' with five red stars. Underneath, there are two buttons: 'Small text' with three stars and 'Large text' with three stars. At the bottom left, a message reads 'Poor contrast for all text sizes. [Click to fix](#)'. On the right, a preview area shows a quote: 'Quote n. 12' followed by the text 'What we think, or what we know, or what we believe is, in the end, of little consequence. The only consequence is what we do.' and the name 'John Ruskin'.

Рис. 1.4. Перевірка на контрастність

Було переглянуто декілька варіантів зображень, на наявність в них альтернативного тексту. Альтернативний текст використовується в двох випадках: коли користувач виконує дії в сервісі зчитування з екрану, або коли через технічні неполадки зображення не відтворюється на сайті. За його допомогою, користувач сервісу зчитування тексту може отримати повний контент на сторінці [3].

Розглянемо зображення на карточках для набору тарифів за споживанням електроенергії (рис. 1.5).

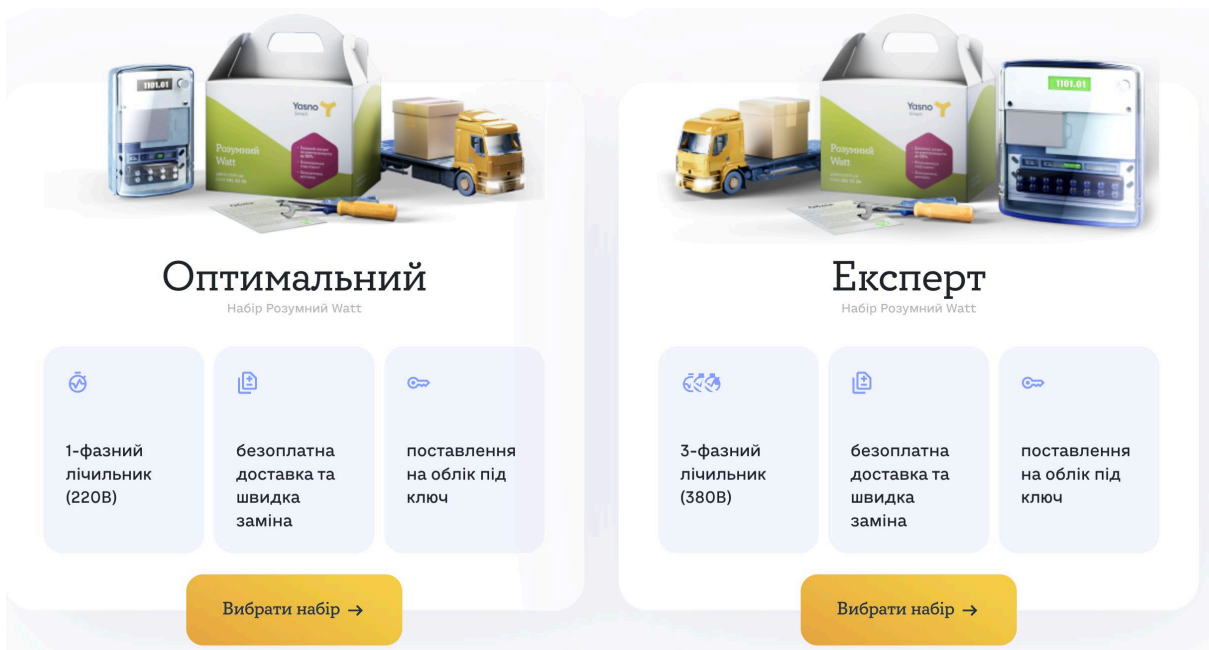


Рис. 1.5. Зображення на карточках для набору тарифів за споживанням електроенергії

Переходячи до відкритого коду за допомогою DevTools знаходимо відповідні зображення та перевіряємо наявність заповнених alt елементів. Але на жаль, на цих зображеннях альтернативний текст не заповнений (рис. 1.6).

```

<!DOCTYPE html>
<html lang="uk">
  <head>
  </head>
  <body screen_capture_injected="true">
    <div id="app">
      <header id="header" class="header home" style>
      </header>
      <div class="hero">
        <div class="hero-container">
          <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 112 34" class="hero-logo">
          </svg>
          <picture class="hero-pic">
            <source srcset="/smart-watt/images/hero-man.webp 1x, /smart-watt/images/hero-man-2x.webp 2x" type="image/webp">
            <img src="" alt="" type="image/png">
          </picture>
        </div>
      </div>
      <div class="sect">
        <div class="co">
          <h2 class="s">
            <h3 class="card-title">Оптимальний</h3>
            <p class="card-description">Наб: https://yasno.com.ua/smart-watt/images/plan-pic-1.png
            <p class="plan-advantages">
              <ul class="plan-advantages">
                <li>1-фазний лічильник (220В)</li>
                <li>безоплатна доставка та швидка заміна</li>
                <li>поставлення на облік під ключ</li>
              </ul>
            <a href="https://shop.yasno.com.ua/shop/sets?utm_source=yasno.com.ua&utm_medium=referral" class="watt-btn" target="_blank">
            </a>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

Рис. 1.6. Перевіряємо наявність заповнених елементів

Як ми бачимо цей сайт є важливою соціальною потребою, особливо коли він надає такі послуги як особистий кабінет, оплата через сайт та інші дії. Але даний сайт не має повної інклюзії та доступності для всіх верст населення.

Ще один із прикладів не повного застосування доступності контенту на веб-сайті – це сайт центрального універмагу в Києві, головний ЦУМ країни [4]. З першого погляду, не можна помітити значних недоліків на сайті. Присутні різні ефекти при наведенні на посилання та зміна курсора зі звичайного на вказівний, що дає користувачеві розуміння того, що це є клікабельний елемент. Наприклад, логотип сайту, який розміщено на шапці сайту, покращує роботу з ним, з його допомогою користувач завжди має доступ до головної сторінки. В звичайному стані він має чорне забарвлення на білому тлі (рис. 1.7), але при наведенні курсору на нього, логотип змінює своє забарвлення в рожевий колір, та відбувається зміна курсору «стрілка» на «вказівний» (рис. 1.8).



Рис. 1.7. Вигляд логотипу сайту в звичайному стані(без взаємодії з курсором)



Рис. 1.8. Ефект при наведенні на логотип сайту

Але прогортавши далі, можна побачити, що в підвалі сайту знехтували рекомендацією щодо контрастності тексту та фону (рис. 1.9).

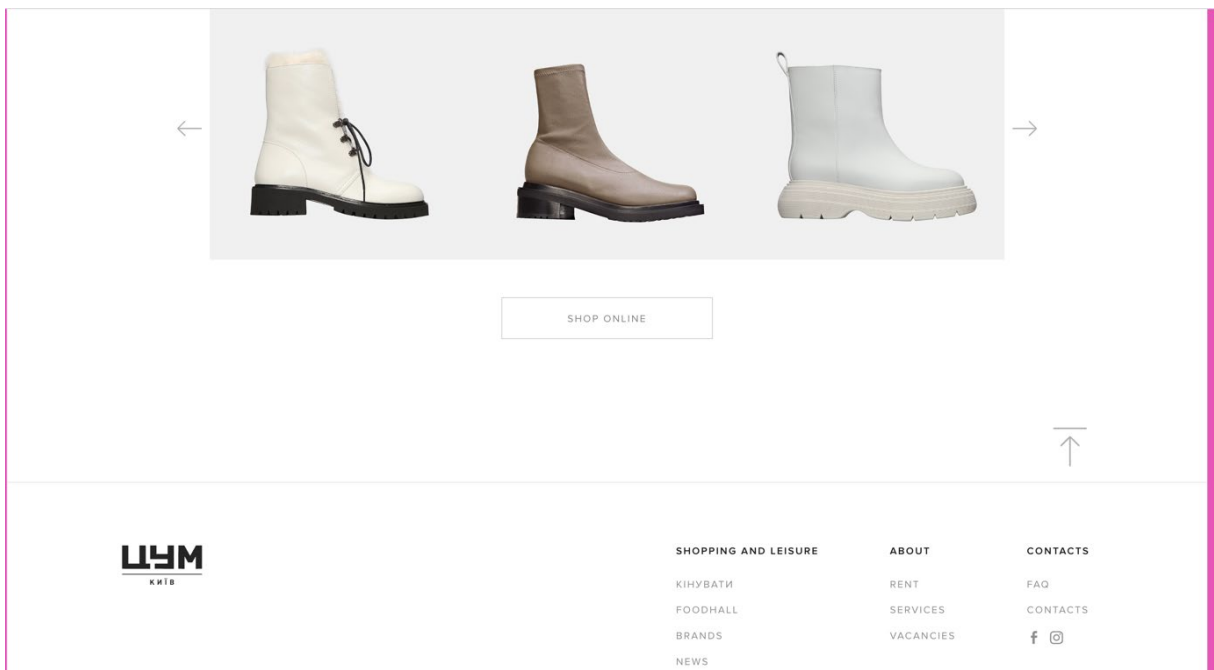


Рис. 1.9. Приклад недостатньої контрастності між фоном та текстом на офіційному сайті центрального універмагу столиці

Інформацію про колір тесту було взято з відкритого коду, через можливості браузера використовувати один з найпотужніших інструментів для розробника та тестувальника веб-проектів, DevTools(developer tools). Протестувала контрастність кольорів на спеціальному сервісі, який використовувався і для попередніх тестувань, та отримала такі результати: контраст дуже слабкий для всіх розмірів тексту (рис. 1.10).

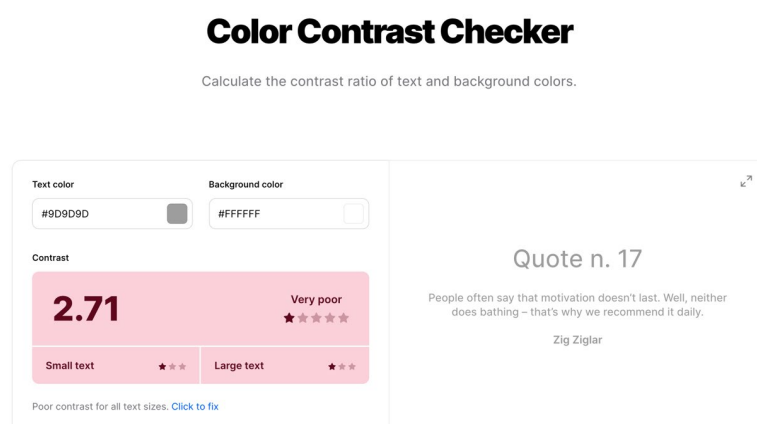


Рис. 1.10. Результати тестування контрастності кольору тексту та фону в підвалі сайту

Також на рисунку 1.9 зображено останній блок сайту що розсташований перед підвалом. Ми можемо побачити, що і цьому блоці зображено ряд товару, який має рекламувати певну продукцію, а під ним одразу кнопка покупки цього товару. Зважаючи на те, що ця кнопка в один колір текстом, що знаходиться в футері, ми отримуємо такий результат:

- Людина з поганим зором втрачає контент і не може так швидко зорієнтуватись та виконати покупку.

- Власник сайту, втрачає прибуток, через погану видимість кнопки.

Тому, можемо зробити висновки, що при плануванні дизайну, клієнт та дизайнер мають погодити основні вимоги до сайту, адже в майбутньому, це може вплинути на різноманітні фактори.

Продовжуючи вивчати головну сторінку сайту, було перевірено наявність альтернативного тексту на каруселі(або слайдер), який рекламує взуття (рис. 1.11).

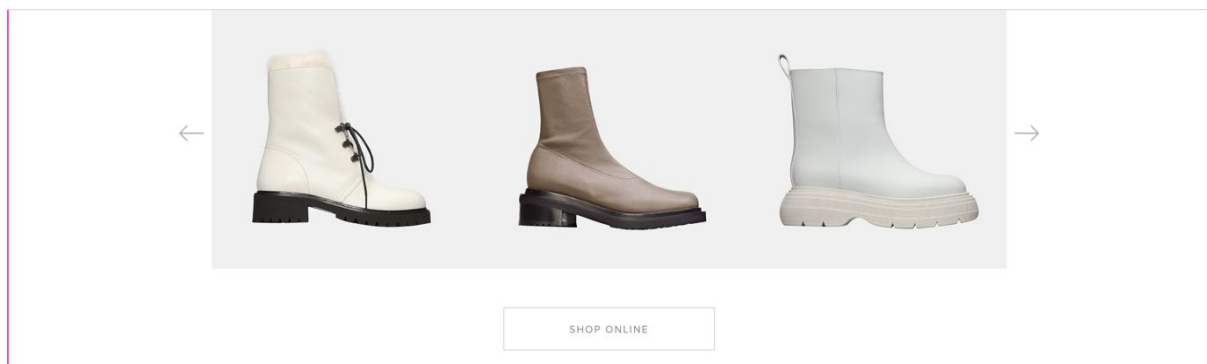


Рис. 1.11. Слайдер на якому зображено взуття певного бренду

Використовуючи DevTools отримала доступ до відкритого коду сайту та перевірила наявність альтернативного тексту на зображенні. Елемент, який відповідає за альтернативний текст, виявився порожнім (рис. 1.12).

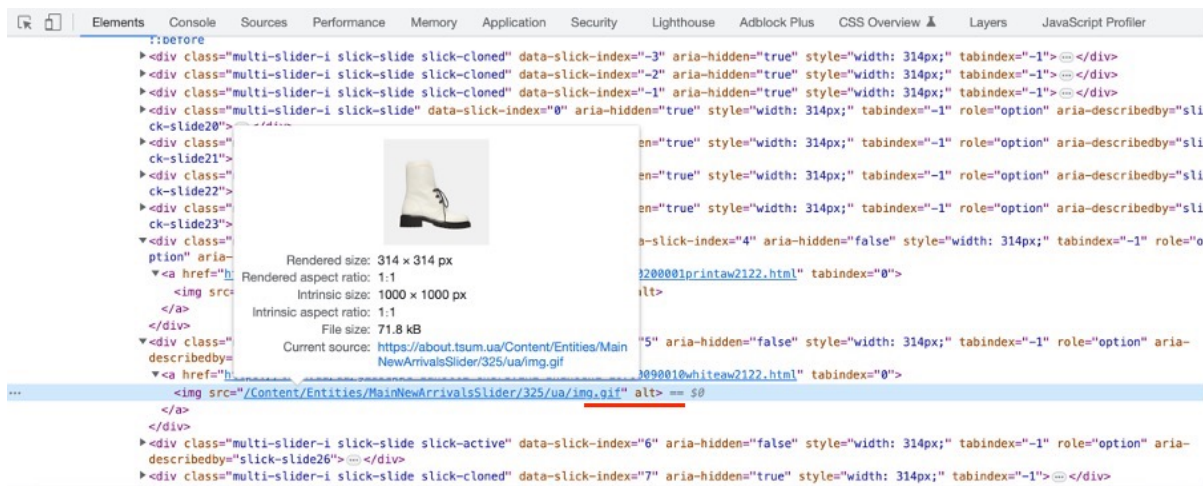


Рис. 1.12. Елемент, який відповідає за альтернативний текст, виявився порожнім

У підсумку, люди, які користуються програмою зчитування з тексту втратять контент через відсутність альтернативного тексту.

Не менш важливо, зробити такий сайт, щоб користувач міг отримувати весь важливий матеріал сайту без мишки/курсора або використання тачпаду. На жаль, на сайті, який я навела до прикладу, це не враховується. Якщо більш детально, перехід по табуляціям відбувається, але без фокусування на посиланні. Єдина підказка, про те що табуляція існує – це інформація від браузера внизу сторінки (рис. 1.13).

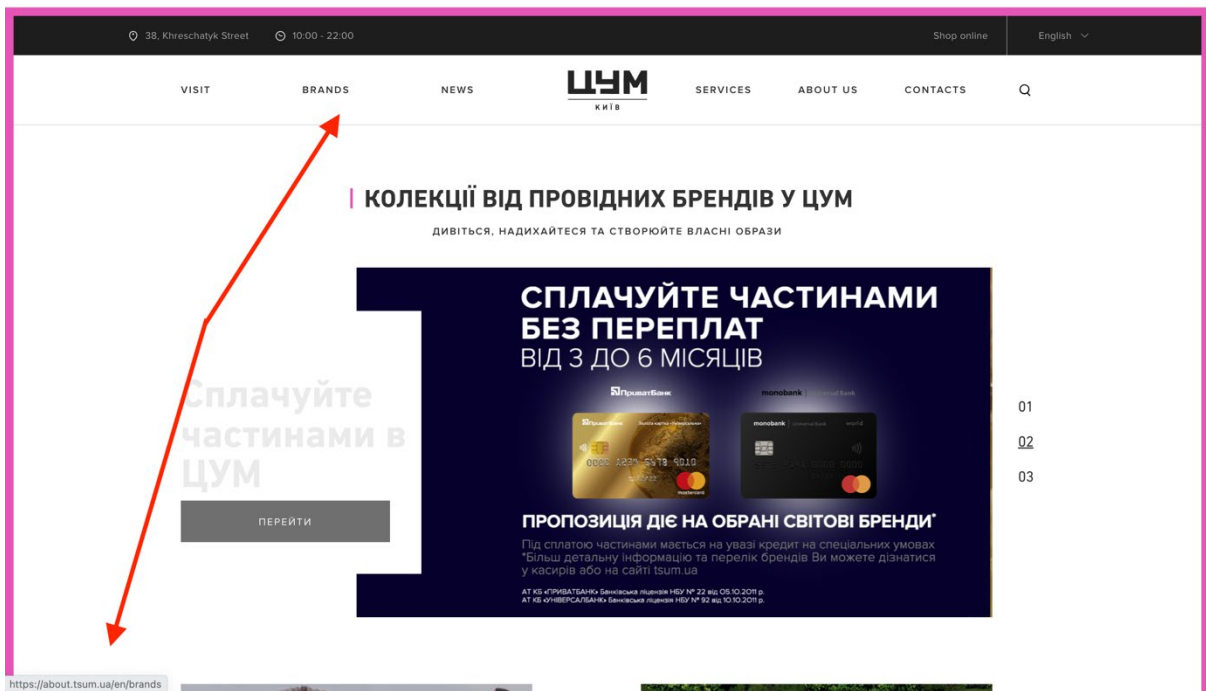


Рис. 1.13. Інформація про посилання без фокусування при використанні табуляції

Переглядаючи сайт за допомогою зчитування з екрана помітила, що зчитувач пропускає в меню лише ті елементи списку, які мають дочірні підпункти. Тобто користувач не отримує багато інформації, що може призвести до непорозуміння або втрати великої частини контенту. Наприклад, на цьому сайті втрачаються такі елементи списку як «Відвідати» та «Про нас» (рис. 1.14).



Рис. 1.14. Елементи списку, які не відтворюються через програму зчитування з екрану.

Розглянемо більш детально кожен із пунктів. Список елементу «Відвідати» має таких чотири підпункти: «Кінувати», «Смакувати», «Акції» та «Події» (рис. 1.15)

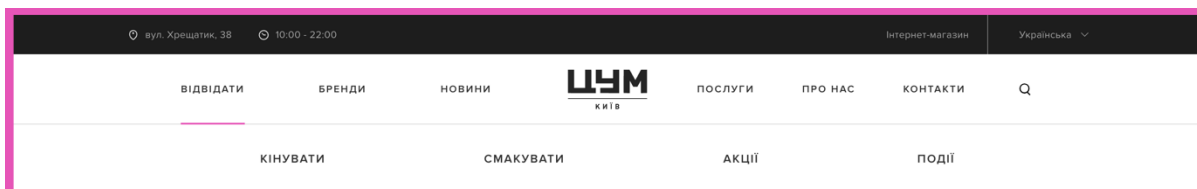


Рис. 1.15. Зображено список елементів, які включає в себе пункт списку «Відвідати»

Якщо я, як користувач, зайшла дізнатись на якому поверсі знаходиться кіно та чи є воно взагалі в універмазі, або вирішила дізнатись, які є місця де можна перекусити, або хочу піти на якусь подію і треба дізнатись час, то я не зможу цього зробити, через втрату контенту.

Проаналізувавши ці сайти, я зробила для себе висновки, що обов'язково маю звертати увагу на контрастність кольорів та на те, як сервіси для зчитування з екрану допомагають користувачам отримувати інформацію.

Переглянувши негативні приклади, які зустрічаються в Інтернеті, я хочу також проаналізувати, доступні та виконані по всім вимогам веб-сайти.

Одним з найпопулярніших інформаційних сайтів у всьому світі є веб-ресурс зі світовими новинами BBC [5]. Тому обравши цей сайт для аналізу доступності, я була впевнена, що він буде ідеальним, адже маючи такий високий рівень популярності нехтування такими значними рекомендаціями щодо доступності могли б призвести до втрати рейтингу та довіри до цієї компанії.

Отже, почнемо розбір відразу двох посилань: головна сторінка [5] та новини українською [6]. Так як дизайн сайтів відрізняється, то було в першу чергу порівнювано на контрастність тексту та фону, а також на роботу по сайту через табуляцію та сервіс зчитування з екрану.

Почнемо огляд сайтів з контрастності між фоном та кольором тексту. Було обрано блоки з головними новинами на основній сторінці. Порівнявши два посилання, можна зробити висновки, що українська версія має чіткіші значення контрастності кольорів. На сторінці сайту переважна більшість тексту в чорному та темно-сірому кольорах на білому тлі (рис. 1.16).

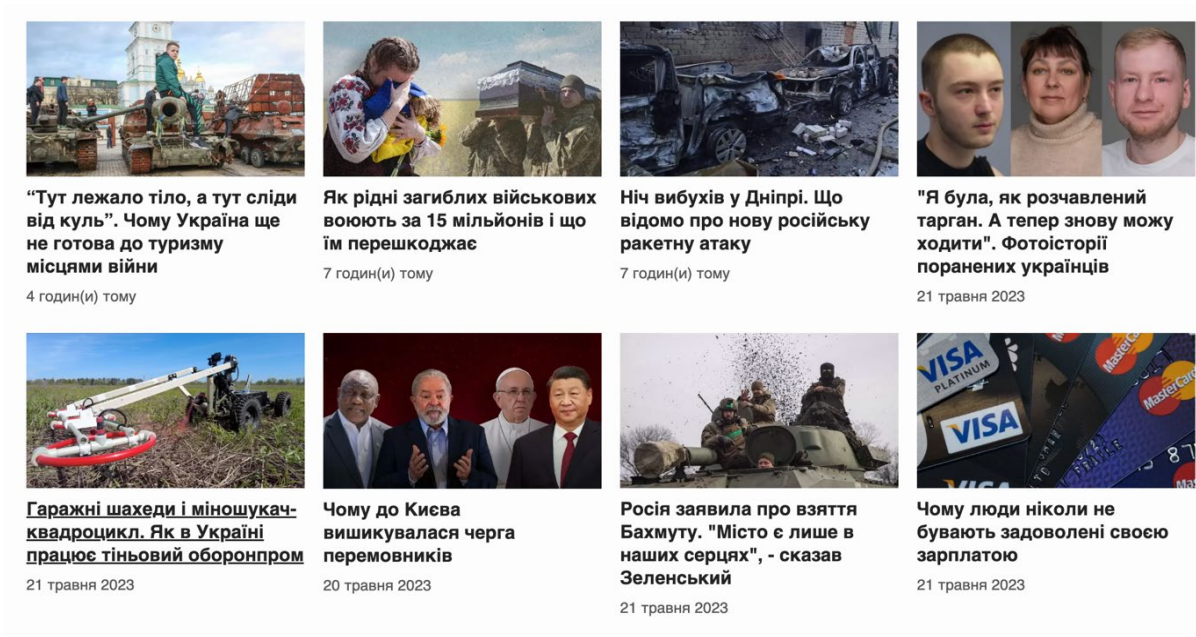


Рис. 1.16. Приклад блоків з головними новинами на українській версії сайту

Для підтвердження даної інформації робимо тести на спеціальному сервісі, який відповідає вимогам по доступності. Для тестування використовуємо поле з датою створення статі (рис. 1.17), так як вона має більш світлий колір.

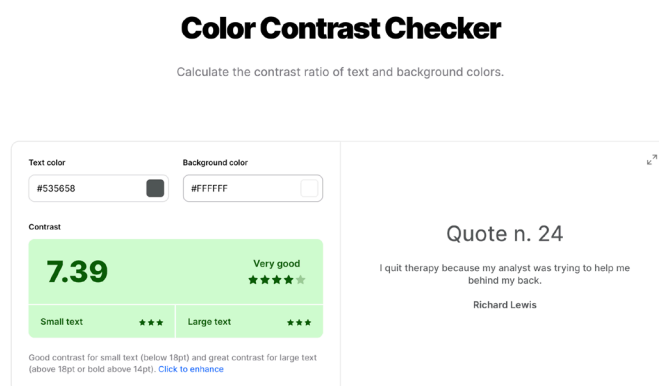


Рис.1.17. Результати тесту контрастності для посилання на україномовної аудиторії

Переглянемо такі ж блоки на англійській та фундаментальній версії сайту. Скріншот взято з блоку з новинами. Але відразу можна помітити, що другорядний текст має світліший колір (рис. 1.18), ніж той, який є на

українській версії сайту (рис. 1.16). Тому, переглянувши зображення, прийнято рішення взяти колір з поля, який відповідає за тег блогу. Отримані результати є в граничній доступності, але не є ідеальними (рис. 1.19).

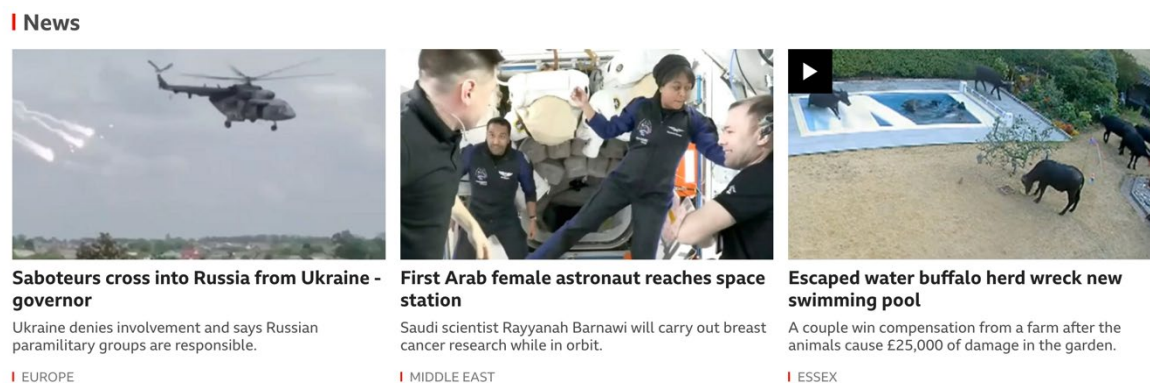


Рис. 1.18. Скріншот з блоку новин англійської версії сайту

Color Contrast Checker

Calculate the contrast ratio of text and background colors.

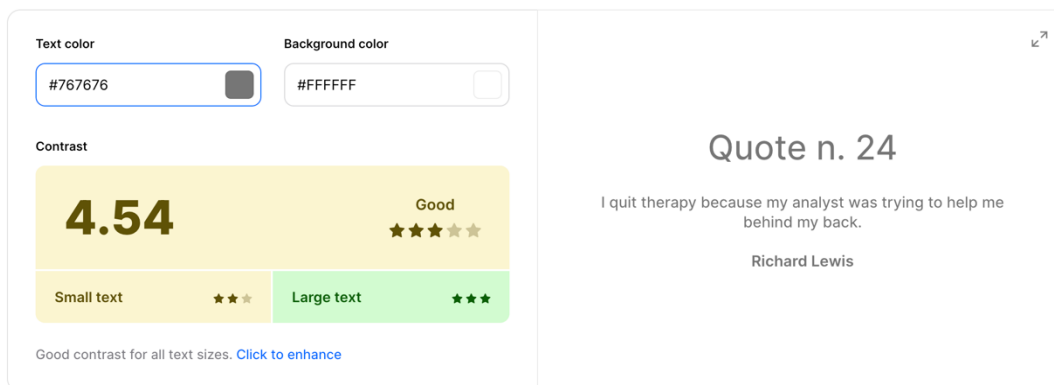


Рис.1.19. Результати тесту контрастності для посилання на англійській мовній аудиторії

Отже, можемо зробити висновки, що за посиланням на українськомовній сторінці контент має кращу контрастність ніж на англійськомовній сторінці. Не менш важливим є альтернативний текст до зображень, що у випадку зчитування з екрану доповнить та дасть більше розуміння контексту та матеріалу, який хоче донести власник сайту.

Для кращого розуміння, візьмемо два повністю однакових зображення (рис. 1.20) (це ми можемо взяти в частині «Новина дня») та порівняємо наявність опису, та його зміст.

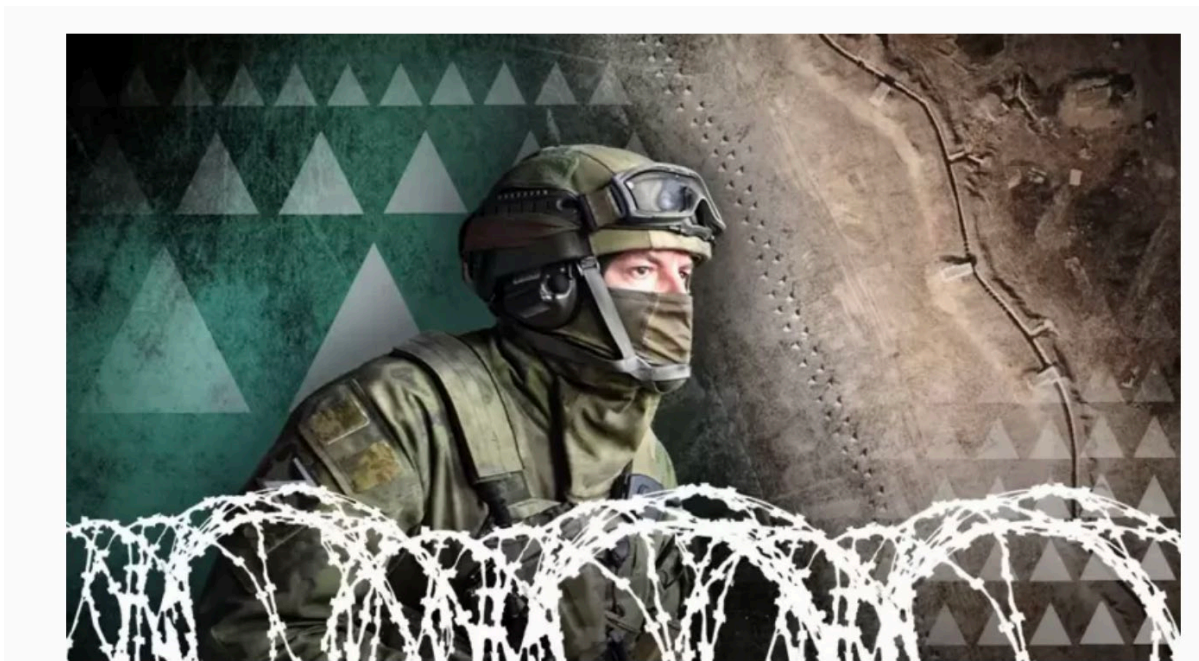


Рис. 1.20. Зображення, яке використовувалось на двох посиланнях

Почнемо огляд з української версії сайту, де альтернативний текст присутній, але він зовсім не описує того, що зображено на картинці, тому не можемо вважати це повноцінним альтернативним текстом (рис. 1.21). Альтернативний текст включає в себе лише одне слово «Війна», але не описує зображення.

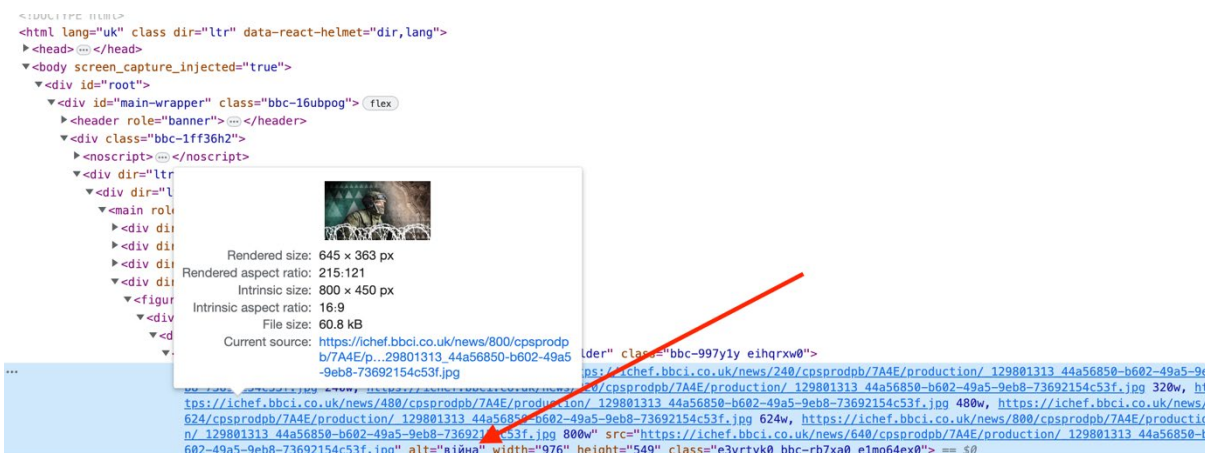


Рис. 1.21. Альтернативний текст на сторінці україномовного контенту

Розглянувши альтернативний текст на сторінці англomовної аудиторії, я помітила що він є більш влучним та доцільним. Адже він описує саме те, що зображено на малюнку (рис. 1.22), а не просто описує тему статі, як показано в минулому прикладі.

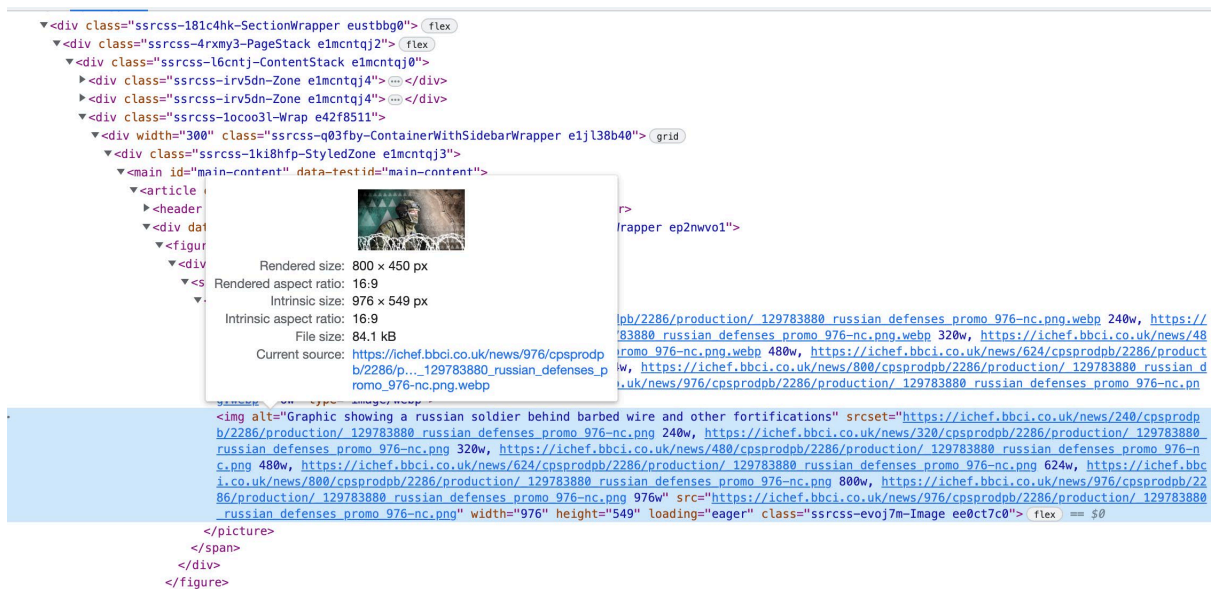


Рис. 1.22. Альтернативний текст на сторінці англomовного контенту

Розглядаючи табуляцію та програму зчитування з тексту я не знайшла жодних недоліків, та взяла це до уваги при створенні власного сайту. На сайтах було враховано табуляцію по всім посиланням, робота з пошуковою системою за рахунок клавіатури, навігація по дочірнім елементам списку, чого не було враховано на сайті універмагу столиці (рис. 1.14).

Отже, розглядаючи різноманітні сайти, можна знайти безліч переваг та недоліків, які використовуються для доступності та інклюзивності сайту. Аналізуючи закордонні сайти роблю висновок, там проблему інклюзивності підіймають та намагаються вирішити, ми побачили приклади з альтернативним текстом для ідентичного зображення (рис. 1.20 та рис. 1.21). В своїй роботі, я прагну досягти максимальної доступності для людей з різними можливостями. Враховуючи переваги та недоліки зробити сайт з контрастністю, альтернативним текстом, вільним пересуванням по сайту без мишки/тачпаду

(лише за допомогою клавіатури) та з використанням зчитування з екрану передати весь зміст сайту.

1.2. Призначення розробки та галузь застосування

Розробка веб-сайту з допоміжними функціями для людей з обмеженими можливостями. Такі сайти можуть використовуватись не тільки людьми з усіма можливостями, включаючи не лише технічні, а і фізичні можливості. Галузь застосування – використання таких технологій та функціоналу на будь-яких сайтах.

1.3. Підстави для розробки

В кінці навчання, студент виконує кваліфікаційну роботу (проект). Тема роботи узгоджується з керівником проекту, випускаючою кафедрою.

Підставою для розробки кваліфікаційної роботи на тему «Розробка веб-сайту для людей з обмеженими можливостями» є наказ по Національному технічному університету «Дніпровська політехніка» від 16.05.2023 № 350-с.

1.4. Постановка завдання

Метою кваліфікаційної роботи є створення веб-сайту для людей з обмеженими можливостями, що забезпечує використання продукту для більш широкої аудиторії та надає доступ людям з обмеженими можливостями до інформації, яка знаходиться на сайті, що впливає на два фактори:

- З боку користувача. Користувач матиме змогу отримати більше інформації з інформаційних джерел в Інтернеті.
- З боку власника сайту. Власник зможе охопити більшу аудиторію користувачів, які б мали доступ до продукту власника, що в свою чергу збільшує конверсію.

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

Процес роботи з адмін-панеллю та редагуванням сайту, має здійснюватись за допомогою введення даних у відповідні поля, та натискання кнопок.

Потрібно передбачити можливість збереження параметрів веб-сайту до бази даних для подальшого виклику з метою прискорення роботи спеціаліста, а також створення послідовності команд.

1.5.2. Вимоги до інформаційної безпеки

На сайті має бути підключений SSL-сертифікат, який передбачає захист даних користувачів сайту.

Також, в разі необхідності, доступу до сторінок сайту через адміністративну панель для кваліфікованих фахівців, потрібно передбачити можливість введення паролю, доступ до виклику якого не має бути явним.

Передбачити та запобігти таким можливим типам вразливості:

- Можливість запису/виконання виконуваних файлів стороннім користувачем
- Впровадження SQL через електронні листи, відповіді та форми
- XSS атаки
- Вразливості зворотного відстеження
- Підробка форми
- Уразливості CSRF

1.5.3. Вимоги до складу та параметрів технічних засобів

Для повноцінного функціонування при умові редагування сайту необхідні:

- персональний комп'ютер;
- клавіатура;

- миша.

Певних вимог до потужності комп'ютера не висувається, оскільки достатньо запуску на ньому операційної системи.

Операційна система яка підтримує PHP7.2 та вище (до 8.2 включно), встановлена СУБД MySQL 8.0 та вище або MariaDB 10.6 та вище. Встановлений веб-сервер Apache 2.4 або Nginx 1.23.

1.5.4. Вимоги до інформаційної та програмної сумісності

Розроблюваний веб-сайт сумісний з браузерами:

- Chrome,
- FireFox,
- Safari,
- Opera,
- Internet Explorer.

Сайт є кроссбраузерним та адаптивним, тому він доступний на різних девайсах:

- мобільні телефони,
- планшети,
- ноутбуки,
- ПК,
- Телевізори.
- Портативні(мобільні) пристрої

Також, доступні такі операційні системи:

- Mac OS,
- Windows OS,
- Linux OS,
- Android,
- iOS.

Додаток має бути розроблений за допомогою HTML-розмітки, CSS, JS, PHP та розміщено на платформі WordPress.

Веб-сайт працює без компіляції та виконується мовами програмування, що інтерпретуються.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

2.1. Функціональне призначення програми

Мета цієї роботи – це створення веб-сайту на платформі WordPress, функціональне призначення програми включає в себе створення та редагування контенту. Програма забезпечує можливість додавання, редагування та видалення різних типів контенту на веб-сайті, таких як статті, сторінки, зображення, відео, аудіо, текст тощо. Це реалізовано за допомогою інтерфейсу для введення тексту, завантаження файлів, встановлення зображень та відео.

Програма забезпечує підтримку плагінів та розширень. Вона має можливість встановлювати та налаштовувати різноманітні плагіни та розширення, які розширяють функціональність веб-сайту. На сайті додано плагін, PageSpeedNinja[6], він покращує показники індексу швидкості завантаження сайту, а також й додає значну кількість покращень, що оцінюються алгоритмами Google PageSpeed Insights[7], який надає звіт про взаємодію користувача зі сторінкою як на мобільних, так і на настільних пристроях, також, надає пропозиції покращення їх.

Реалізація зручного користування адмін-панелі є важливим питанням у розробці продукту, тому було прийнято рішення використовувати плагін ACF[8] (Advanced Custom Fields), які мають зручний інтерфейс для адміністратора та зменшують час роботи над додаванням, редагуванням або видаленням певних форм та полей на сторінках сайту.

2.2. Опис застосованих математичних методів

В даній роботі було використано математичні методи для перевірки валідності введених користувачем даних. Для реалізації перевірки валідності було використано вбудовані функції мови програмування JavaScript [9].

2.3. Опис використаної архітектури та шаблонів проектування

При інтеграції форм відправки повідомлень з CRM HubSpot був вибір використання між двох шаблонів(патерн): MVC [10] (Model-View-Controller) та MVVM (Model-View-ViewModel).

Провівши дослідження та проаналізувавши обидва шаблони зробила такі висновки: найбільш стандартним способом підключення моделі даних до представлення програми є інтерфейс, який називається контролером. У шаблоні MVC контролер діє як інструмент, який безпосередньо маніпулює даними заданої моделі, а в шаблоні MVVM, на відміну від методу контролера, метод ViewModel залежить від зовнішнього інтерфейсу програми. MVC краще ніж MVVM тим, що модель включається один раз, і фактично ми можемо бачити усі звернення модулю до бази даних в одному місці. З урахування MVVM моделі нам довелося б використовувати звернення до БД ще й у шаблоні, що могло призвести до помилок рендеру за людським фактором, окрім того, у деяких випадках це б могло відкрити вразливість до SQL-ін'єкцій. В шаблоні MVC контролер та модель розміщуються в папці плагіну wp-content/plugins

Саме тому в цьому випадку найкращим рішенням було використання шаблону MVC , що розділяє архітектуру проекту на три модулі відповідно: модель, подання та контролер. Він дозволяє підтримувати зміни кожного компонента в незалежності один від одного. Логічна схема роботи шаблону зображено на рисунку 2.1.

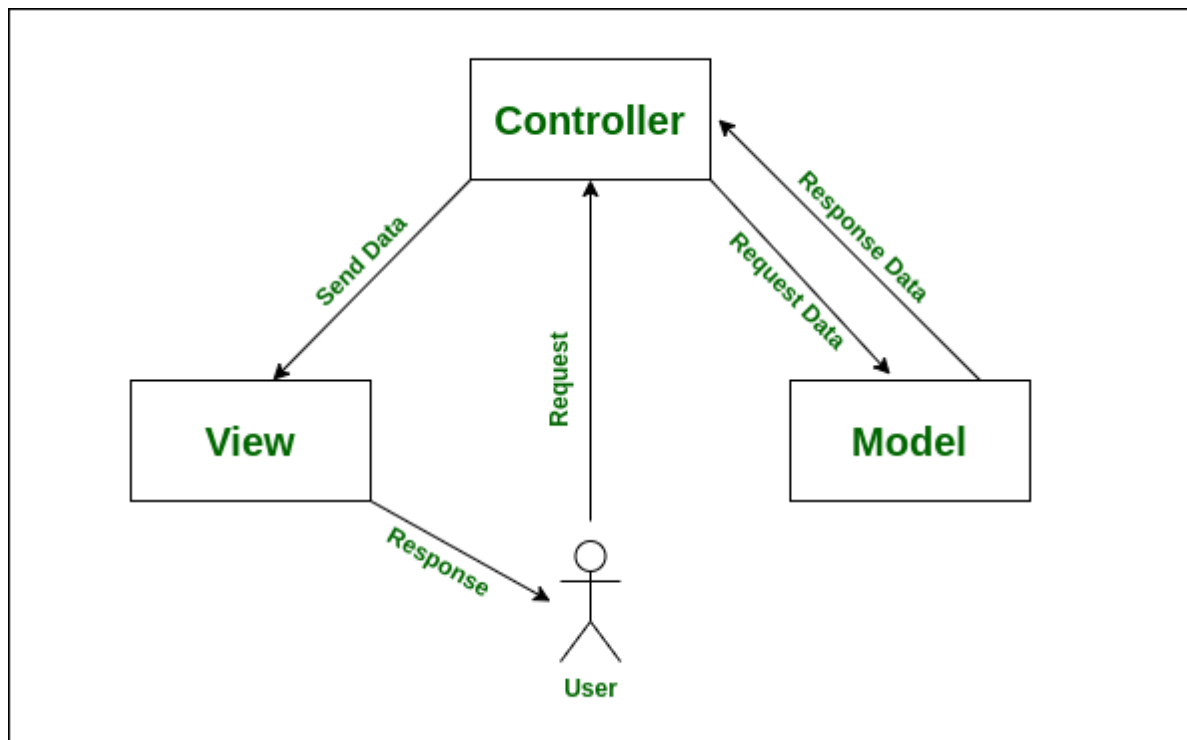


Рис. 2.1. Логічна схема роботи шаблону MVC

Робота MVC розробленого продукту є подібною до тої, яку наведено нижче. Більш детальніше розглянемо на рисунку 2.2. Після HTTP-запита від користувача сервер розрізняє його на методи GET та POST. Якщо використовується GET запит, тобто все що можна написати в пошуковому рядку, як наприклад, можливість переключатись на іншу сторінку, за допомогою рядка, змінивши назву поточної сторінки на іншу. В цьому випадку йде лише рендер інформації для виведення користувачеві, іншої глобальної роботи з БД не виконується. Якщо детальніше, не ведеться запис даних, не перезаписуються дані, не надсилаються на пошту адміністратора і т д.

Коли користувач, вводить дані через метод POST, а це є варіантом заповнення контактної форми, йде більш обширина робота з базою даних з метою запису інформації, надсилання e-mail та виконання будь-яких глобальних змін в стані програми.

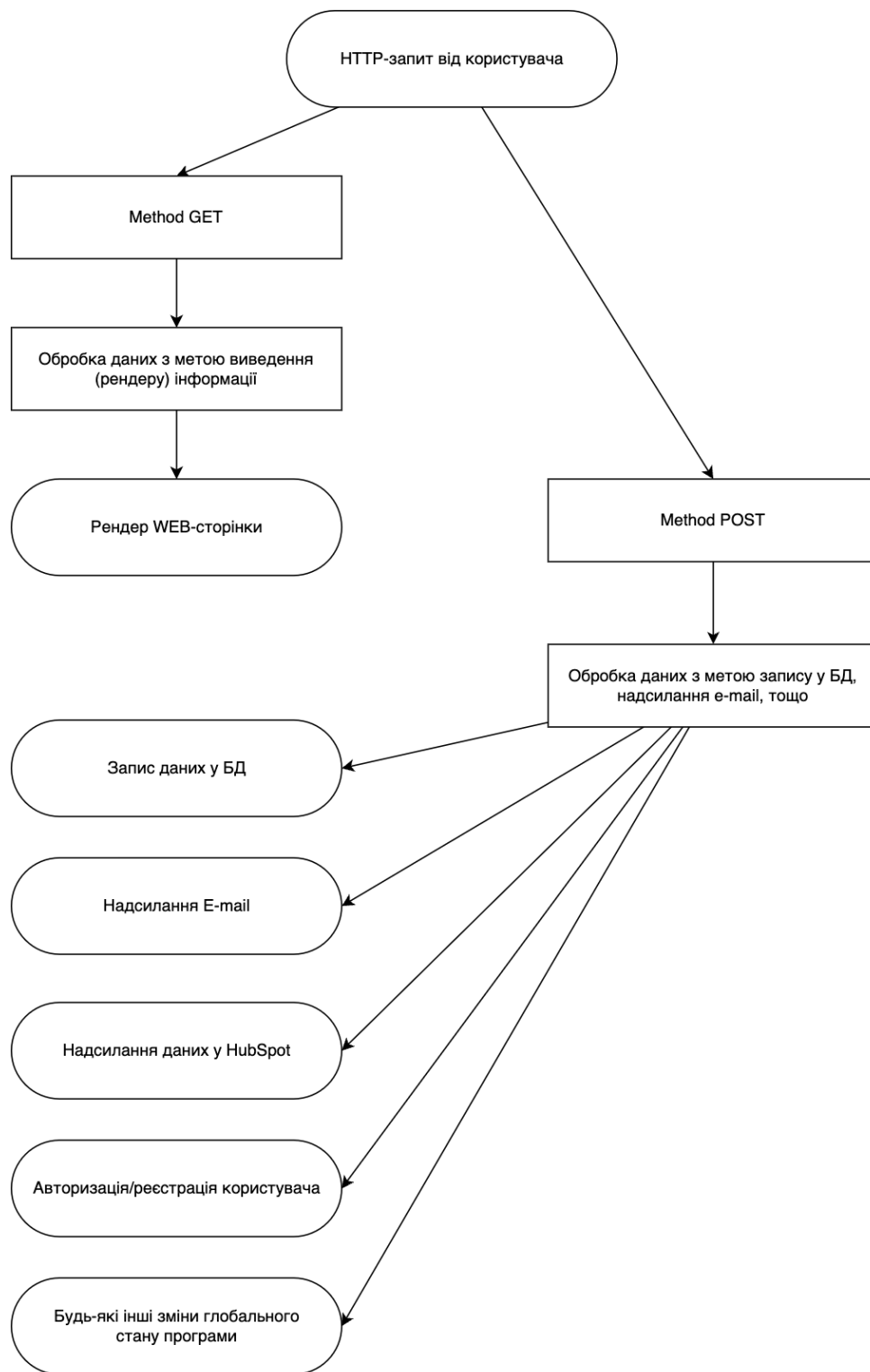


Рис. 2.2. Зображення блок-схеми роботи MVC в розрбленому продукту

В роботі використовувалась стандартна база даних CMS WordPress для внесення даних, а також, їх доробок було застосовано, за рекомендацією кодексу WordPress[11] (Кодекс WordPress: Кодекс – це енциклопедія всього, що

стосується WordPress. Це найповніше джерело інформації про WordPress.), в переважній більшості випадків використовувалося зберігання Json в осередках зберігання бази даних. Нормалізація відношень БД[12] не проводилась, так як WordPress працює з ненормалізованою БД, що має деякі переваги в контексті WordPress у порівнянні з нормалізованою базою даних.

Ненормалізована БД може бути швидшою та легшою для розробки, оскільки не потребує складних структур даних, зв'язків та обмежень, що притаманні нормалізації. Це може пришвидшити процес розробки та розгортання веб-сайту на WordPress. А також ця БД може бути простішою для розуміння та керування даними. Вона може мати більш інтуїтивний структуру, що полегшує роботу з даними та виконання запитів. Особливо в контексті WordPress, де ненормалізована БД може дозволити прямий доступ до даних без потреби складних JOIN-операцій. Завдяки простоті структури та прямому доступу до даних, ненормалізована БД може забезпечити кращу продуктивність при виконанні запитів. Запити до БД можуть виконуватися швидше, оскільки немає необхідності виконувати складні з'єднання між таблицями. БД може дати більшу гнучкість в процесі розробки, що надає можливість швидко вносити зміни до структури БД, додавати нові поля або таблиці без значного впливу на існуючу структуру даних. Це може бути корисним у випадках, коли потрібно швидко змінювати функціональність веб-сайту.

Було враховано, що ненормалізована БД також може мати деякі недоліки, такі як дублювання даних та можливість виникнення неузгодженостей у даних. Тому, при порівнянні ненормалізованої та нормалізованої БД та огляд рекомендацій в застосуванні було однозначно прийняте рішення використовувати ненормалізовану базу даних, так як, якщо б було прийнято рішення нормалізації бази даних, це було б втручанням в ядро CMS (content management system) WordPress, що в подальшому б призвело до різноманітних проблем.

Розглянемо більш детально таблицю "wp-postmeta" та "wp-posts". Таблиця "wp-postmeta" є додатковою таблицею у базі даних WordPress, яка

зберігає додаткові метадані (ключ-значення) для кожного поста. Ці дані дозволяють розширити і налаштувати функціональність поста і забезпечують гнучкість при використанні WordPress.

Таблиця 2.1

Прототип таблиці «wp- postmeta»

Meta_id	Post_id	Meta_key	Meta_value
Це унікальний ідентифікатор для кожного рядка метаданих. Використовується для однозначної ідентифікації запису в таблиці.	Це ідентифікатор поста, до якого відноситься це метадане. Він відповідає ідентифікатору поста в таблиці "wp-posts".	Це назва ключа метаданих, що зберігається. Він використовується для ідентифікації конкретного метаданого.	Це значення, яке зберігається для даного метаданого. Це може бути рядком, числом, масивом або іншим типом даних.

Нижче наведено скріншот таблиці з бд “wp-postmeta” (рис. 2.3).

<input type="checkbox"/> Modify	meta_id	post_id	meta_key	meta_value
<input type="checkbox"/> edit	645	108	services_list_0_list_0_service	Design for Inclusion (DFI)
<input type="checkbox"/> edit	761	182	services_list_0_list_0_service	Design for Inclusion (DFI)
<input type="checkbox"/> edit	879	184	services_list_0_list_0_service	Design for Inclusion (DFI)
<input type="checkbox"/> edit	1037	186	services_list_0_list_0_service	Design for Inclusion (DFI)
<input type="checkbox"/> edit	1223	190	services_list_0_list_0_service	Design for Inclusion (DFI)
<input type="checkbox"/> edit	1427	193	services_list_0_list_0_service	Design for Inclusion (DFI)
<input type="checkbox"/> edit	1631	199	services_list_0_list_0_service	Design for Inclusion (DFI)
<input type="checkbox"/> edit	1875	212	services_list_0_list_0_service	Design for Inclusion (DFI)
<input type="checkbox"/> edit	2095	213	services_list_0_list_0_service	Design for Inclusion (DFI)
<input type="checkbox"/> edit	2324	215	services_list_0_list_0_service	Design for Inclusion (DFI)
<input type="checkbox"/> edit	2549	218	services_list_0_list_0_service	Design for Inclusion (DFI)
<input type="checkbox"/> edit	3508	120	tabs_0_services_list_0_title	Design for Inclusion (DFI)
<input type="checkbox"/> edit	3651	363	tabs_0_services_list_0_title	Design for Inclusion (DFI)

Рис. 2.3. Таблиця “wp-postmeta”

Таблиця "wp-posts" містить основну інформацію про кожний пост в WordPress. Вона зберігає дані, такі як заголовок поста, вміст, автор, дата

публікації, тип поста (стаття, сторінка, прикріплений файл і т.д.), статус публікації (опубліковано, чернетка, архів і т.д.) та інші атрибути, що визначають пост, як приклад було наведено нижче прототип таблиці 2.2. Ця таблиця є основою для управління та відображенням постів в адміністративній панелі WordPress та на веб-сайті.

Таблиця 2.2

Прототип таблиці «wp-posts»

ID	Post-date	Post-title	Post-status	Post-name	Post-type
Це унікальний ідентифікатор для кожного поста. Використовується для однозначної ідентифікації поста в базі даних.	Це дата та час створення поста.	Це заголовок поста.	Це статус поста, який визначає, чи є пост опублікованим, чернеткою, архівом або іншим.	Це URL-френдлі-назва поста, яка використовується в URL-адресі сторінки поста.	Це тип поста, який визначає, чи є пост статтею, сторінкою, прикріпленим файлом або іншим типом контенту.

Прототип таблиці було змальовано з таблиці на рис. 2.4 та спрощено за для більш комфортного читання та розуміння.

<input type="checkbox"/> #	ID	post_author	post_date	post_date_gmt	post_content	post_title	post_excerpt	post_status	comment_status	ping_status	post_password	post_name	to_ping	pinged	post_modified	post_modified_gmt	post_content_filtered	post_parent	guid	menu_order	post_type	post_mime_type	comment_count
<input type="checkbox"/> edit	1015	1	2002-10-17 10:48:05	2002-10-17 10:48:05		Design for Inclusion (DFI)		publish	closed	closed		design-for-inclusion			2002-10-21 18:55:01	2002-10-21 18:55:01	0		https://www.fearlessidures.org/?page_id=1015	0	page		0
<input type="checkbox"/> edit	1016	1	2002-10-17 10:48:05	2002-10-17 10:48:05		Design for Inclusion (DFI)		inherit	closed	closed		1015-revision-v1			2002-10-17 10:48:05	2002-10-17 10:48:05	1015	https://www.fearlessidures.org/?p=1016	0	revision		0	
<input type="checkbox"/> edit	1017	1	2002-10-17 10:48:25	2002-10-17 10:48:25		Design for Inclusion (DFI)		inherit	closed	closed		1015-revision-v1			2002-10-17 10:48:25	2002-10-17 10:48:25	1015	https://www.fearlessidures.org/?p=1017	0	revision		0	
<input type="checkbox"/> edit	1018	1	2002-10-17 10:48:38	2002-10-17 10:48:38		Design for Inclusion (DFI)		inherit	closed	closed		1015-revision-v1			2002-10-17 10:48:38	2002-10-17 10:48:38	1015	https://www.fearlessidures.org/?p=1018	0	revision		0	
<input type="checkbox"/> edit	1055	1	2002-10-18 02:07:28	2002-10-18 02:07:28		Design for Inclusion (DFI)		inherit	closed	closed		1015-revision-v1			2002-10-18 02:07:28	2002-10-18 02:07:28	1015	https://www.fearlessidures.org/?p=1055	0	revision		0	
<input type="checkbox"/> edit	1057	1	2002-10-18 02:12:42	2002-10-18 02:12:42		Design for Inclusion (DFI)		inherit	closed	closed		1015-revision-v1			2002-10-18 02:12:42	2002-10-18 02:12:42	1015	https://www.fearlessidures.org/?p=1057	0	revision		0	
<input type="checkbox"/> edit	1058	1	2002-10-18 02:13:08	2002-10-18 02:13:08		Design for Inclusion (DFI)		inherit	closed	closed		1015-revision-v1			2002-10-18 02:13:08	2002-10-18 02:13:08	1015	https://www.fearlessidures.org/?p=1058	0	revision		0	
<input type="checkbox"/> edit	1059	1	2002-10-18 03:15:23	2002-10-18 03:15:23		Design for Inclusion (DFI)		inherit	closed	closed		1015-revision-v1			2002-10-18 03:15:23	2002-10-18 03:15:23	1015	https://www.fearlessidures.org/?p=1059	0	revision		0	
<input type="checkbox"/> edit	1442	1	2002-10-12 11:06:05	2002-10-12 11:06:05		Design for Inclusion (DFI)		inherit	closed	closed		1015-revision-v1			2002-10-12 11:06:05	2002-10-12 11:06:05	1015	https://www.fearlessidures.org/?p=1442	0	revision		0	
<input type="checkbox"/> edit	1611	1	2002-10-21 18:55:01	2002-10-21 18:55:01		Design for Inclusion (DFI)		inherit	closed	closed		1015-revision-v1			2002-10-21 18:55:01	2002-10-21 18:55:01	1015	https://www.fearlessidures.org/?p=1611	0	revision		0	

Рис. 2.4. Приклад таблиці "wp-posts"

Ці таблиці взаємодіють між собою, таблиця "wp-posts" зберігає основні дані поста, а таблиця "wp-postmeta" містить додаткові метадані і відноситься до

поста за допомогою ідентифікатора поста (post ID), який є зовнішнім ключем для зв'язку між цими таблицями. Ця структура даних дозволяє WordPress ефективно зберігати і керувати різноманітною інформацією, пов'язаною з кожним постом, і забезпечує гнучкість при розширенні функціональності сайту на платформі WordPress.

2.4. Опис використаних технологій та мов програмування

У цьому розділі кваліфікаційної роботи ми детально розглянемо технології та мови програмування, які були використані при розробці веб-сайту для людей з обмеженими можливостями на платформі WordPress. Описані технології та мови дозволяють забезпечити доступність та функціональність сайту для різних категорій користувачів.

Один из основних “кітів” на яких будується сайт це гіпертекстова розмітка HTML[13] (HyperText Markup Language). Він використовується для створення структури та розмітки веб-сторінок. З його допомогою були побудовані основні елементи сайту, включаючи заголовки, параграфи, таблиці, форми та інші важливі компоненти. Наступним елементом являється CSS (Cascading Style Sheets), який використовується для стилізації та оформлення веб-сторінок. Застосування CSS дозволило налаштувати зовнішній вигляд сайту, включаючи кольори, шрифти, розташування елементів та анімацію.

При верстці сайту використовувалась збірка Pug (раніше відомий як Jade) є шаблонізатором HTML, який дозволяє писати більш компактний та зрозумілий код. Використання Pug допомогло нам зменшити кількість коду та полегшити розробку. При збірці проекту, для поліпшення продуктивності та завантаження сторінок, було використано мініфікацію стилів CSS та JavaScript. Це дозволяє зменшити розмір файлів та покращити швидкість завантаження сторінок.

Однієї із найпопулярніших мов програмування веб сайтів є JavaScript[14], а також його бібліотека jQuery, яка дозволяє надавати інтерактивність та

динамічні функціональність на веб-сайті. З його допомогою було реалізовано такі функції як валідація форм, розкриваюче меню, кастомний скрол, а також анімації GSAP[15].

Валідація є невід'ємною частиною в роботі контактної форми, яка була реалізована саме через мову JS. Перевірка електронних адрес допомагає гарантувати, що користувачі вводять правильні електронні адреси. Це дозволяє уникнути передачі недійсних або неправильних адрес та запобігає недоставленню повідомлень до неправильних адресатів. В результаті, маємо забезпечення точність та ефективність комунікації між користувачами та адміністратором сайту.

Якщо користувач не проходить валідацію, яка існує в двох типах: пусте поле форми та неправильно записаний поштовий адрес, то він отримує миттєве (динамічне) повідомлення про помилку, як зображено на рисунку 2.5 (Доповнення: на впливаючому повідомленні написано “Будь ласка, додайте ‘@’ до адресу електронної пошти. ‘Zhivnir.K.O.mnu’ не має ‘@’. ”). А також, повідомлення має автоматично вибирати мову, яка налаштована у браузері користувача) та блокує функцію відправки повідомлення, яка здійснюється через метод POST, а не GET метод, так як він є набагато вразливішим та не є надійним при передачі даних.

The image shows a contact form titled "Complete the form to get in touch with us!". It contains three input fields: "Your name" with the value "Karina", "Company name" with the value "Zhovnir 121-19-1", and "E-mail" with the value "Zhovnir.K.O.mnu". A red border highlights the email field, and a tooltip message reads: "Please include an '@' in the email address. 'Zhovnir.K.O.mnu' is missing an '@'." Below the form, there is a checked checkbox for "I agree to company terms and conditions." and a "Receive a consultation" button.

Рис. 2.5. Зоображено приклад роботи валідації, коли в полі електронної пошти виникла помилка.

Вище було наведено захист на частині фронт-енду, коли йде помилка введенні в полі форми, але якщо, продвинутий користувач, заїде в DevTools, або відключить JS скрипти, то відправка форми відбудеться. Для цього випадку, ми використовуємо бек-енд частину, що при переході інформації на сервер перевіряє дані, і лише потім перенаправляє в кінцевий результат.

Мова PHP (Hypertext Preprocessor) є серверною мовою програмування, яка використовується для обробки даних на сервері та взаємодії з базою даних. Використання PHP дозволило нам інтегрувати верстку сайту в систему WordPress.

MySQL[16] є реляційною системою управління базами даних (СУБД), яка використовується для зберігання та керування даними. Ми використовували

MySQL для збереження інформації, такої як дані про користувачів, вміст сторінок та інші динамічні дані.

Також, у проекті були використані різні плагіни WordPress для полегшення розробки та покращення функціональності. До прикладу, ACF (Advanced Custom Fields) використовувався для створення додаткових полів у публікаціях та дописах, Custom Post Type UI дозволив нам створювати власні типи дописів, що і було однією з наших задач, Duplicate Page спрощував копіювання сторінок це дозволяє адміністратору пришвидшити, полегшити та зберегти свій час над роботою по наповненню контентом, Really Simple SSL допомагає забезпечити безпеку сайту. Він спрощує перехід зі стандартного HTTP на захищений протокол HTTPS[17]. Він автоматично визначає наявність сертифіката SSL на сервері та налаштовує сайт на використання HTTPS.

PageSpeed Ninja сприяв оптимізації швидкості завантаження сторінок, покращує показники Google PageSpeed Insights. Швидкість завантаження сторінок має великий вплив на задоволення користувачів. Веб-сторінки, які завантажуються швидко, забезпечують більш позитивний користувацький досвід та знижують вірогідність відхилень користувачів. А також, Google враховує швидкість завантаження сторінок у своїх алгоритмах ранжування. Якщо веб-сайт завантажується повільно, це може вплинути на його позицію в пошукових системах. Оптимізація швидкості завантаження допомагає поліпшити SEO-показники та позицію в результатах пошуку.

Загалом, оптимізація швидкості завантаження сторінок, що впливає на показники Google PageSpeed, має велике значення для забезпечення високоякісного користувацького досвіду, покращення показників конверсій та поліпшення SEO-показників веб-сайту.

Ці технології та мови програмування були використані у нашому проекті для створення функціонального, доступного та зручного веб-сайту для людей з обмеженими можливостями. Вони дозволили нам забезпечити високу якість веб-додатка та надати користувачам зручний інтерфейс для взаємодії з сайтом.

Використання бази даних в WordPress — це набір таблиць (у коробковій версії цієї таблиці 12 штук). Кожна таблиця має унікальну назву, яка починається з префікса бази даних. За змочуванням префікс БД — wp (рис. 2.6)

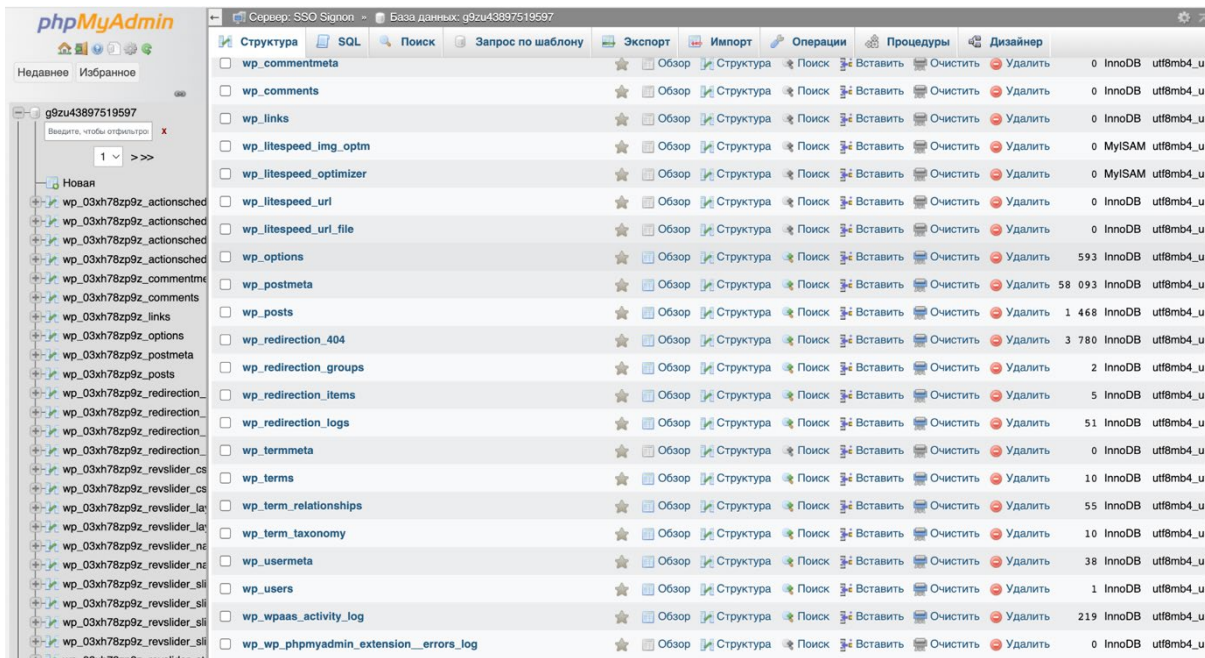


Рис. 2.6 Приклад префіксу за змочуванням

На скріншоті (рис. 2.7) нижче представлений візуальний огляд бази даних WordPress та зв'язків між таблицями, створеними під час стандартної установки WordPress.

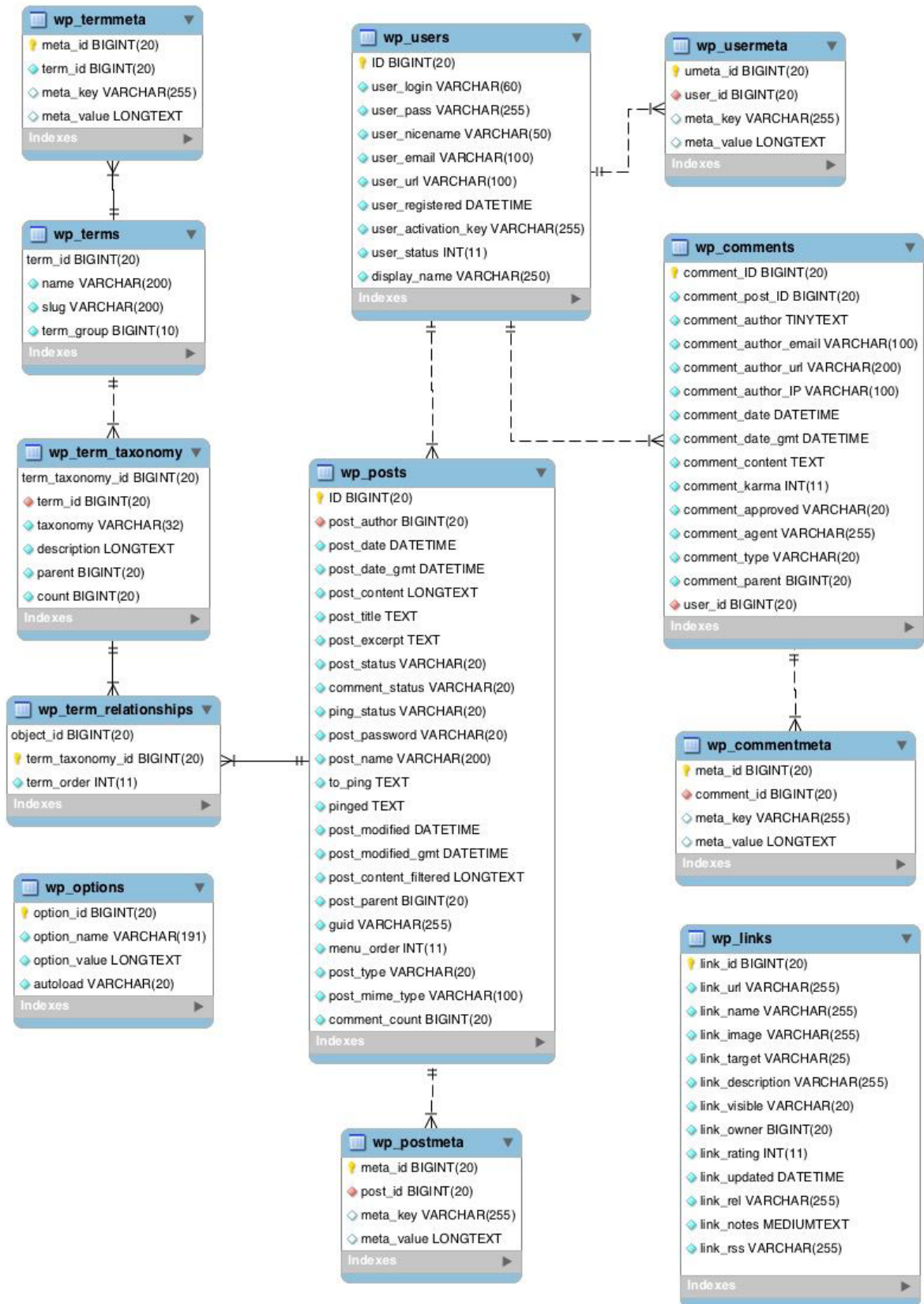


Рис. 2.7. Візуальний огляд бази даних WordPress та зв'язків між таблицями

Також, при експортуванні таблиці вона має такий вигляд (рис. 2.8).


```

64537
64538 CREATE TABLE `wp_termmeta` (
64539   `meta_id` bigint(20) UNSIGNED NOT NULL,
64540   `term_id` bigint(20) UNSIGNED NOT NULL DEFAULT '0',
64541   `meta_key` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
64542   `meta_value` longtext COLLATE utf8mb4_unicode_ci
64543 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
64544
64545 -----
64546
64547 ---
64548 --- Структура таблиці `wp_terms`
64549 ---
64550
64551 CREATE TABLE `wp_terms` (
64552   `term_id` bigint(20) UNSIGNED NOT NULL,
64553   `name` varchar(200) COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT '',
64554   `slug` varchar(200) COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT '',
64555   `term_group` bigint(10) NOT NULL DEFAULT '0'
64556 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
64557
64558 ---
64559 --- Дамп даних таблиці `wp_terms`
64560 ---
64561
64562 INSERT INTO `wp_terms` (`term_id`, `name`, `slug`, `term_group`) VALUES
64563 (1, 'Uncategorized', 'uncategorized', 0),
64564 (2, 'fearless', 'fearless', 0),
64565 (3, 'text', 'text', 0),
64566 (4, 'after_paragraph', 'after_paragraph', 0),
64567 (5, 'sample', 'sample', 0),
64568 (6, 'message', 'message', 0),
64569 (7, 'php', 'php', 0),
64570 (8, 'everywhere', 'everywhere', 0),
64571 (9, 'disable', 'disable', 0),
64572 (10, 'comments', 'comments', 0);
64573
64574 -----
64575
64576 ---
64577 --- Структура таблиці `wp_term_relationships`
64578 ---
64579
64580 CREATE TABLE `wp_term_relationships` (
64581   `object_id` bigint(20) UNSIGNED NOT NULL DEFAULT '0',
64582   `term_taxonomy_id` bigint(20) UNSIGNED NOT NULL DEFAULT '0',
64583   `term_order` int(11) NOT NULL DEFAULT '0'
64584 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
64585

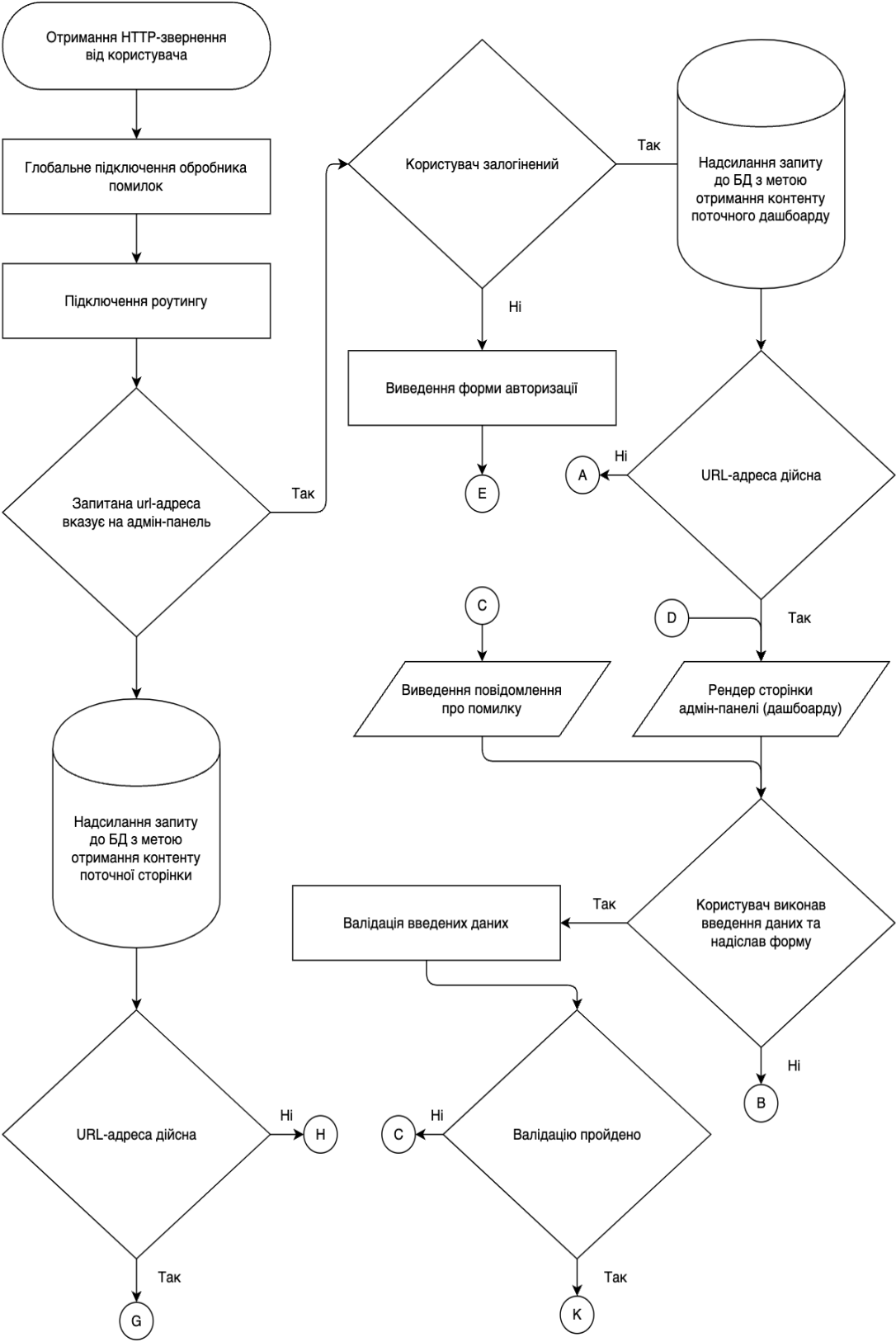
```

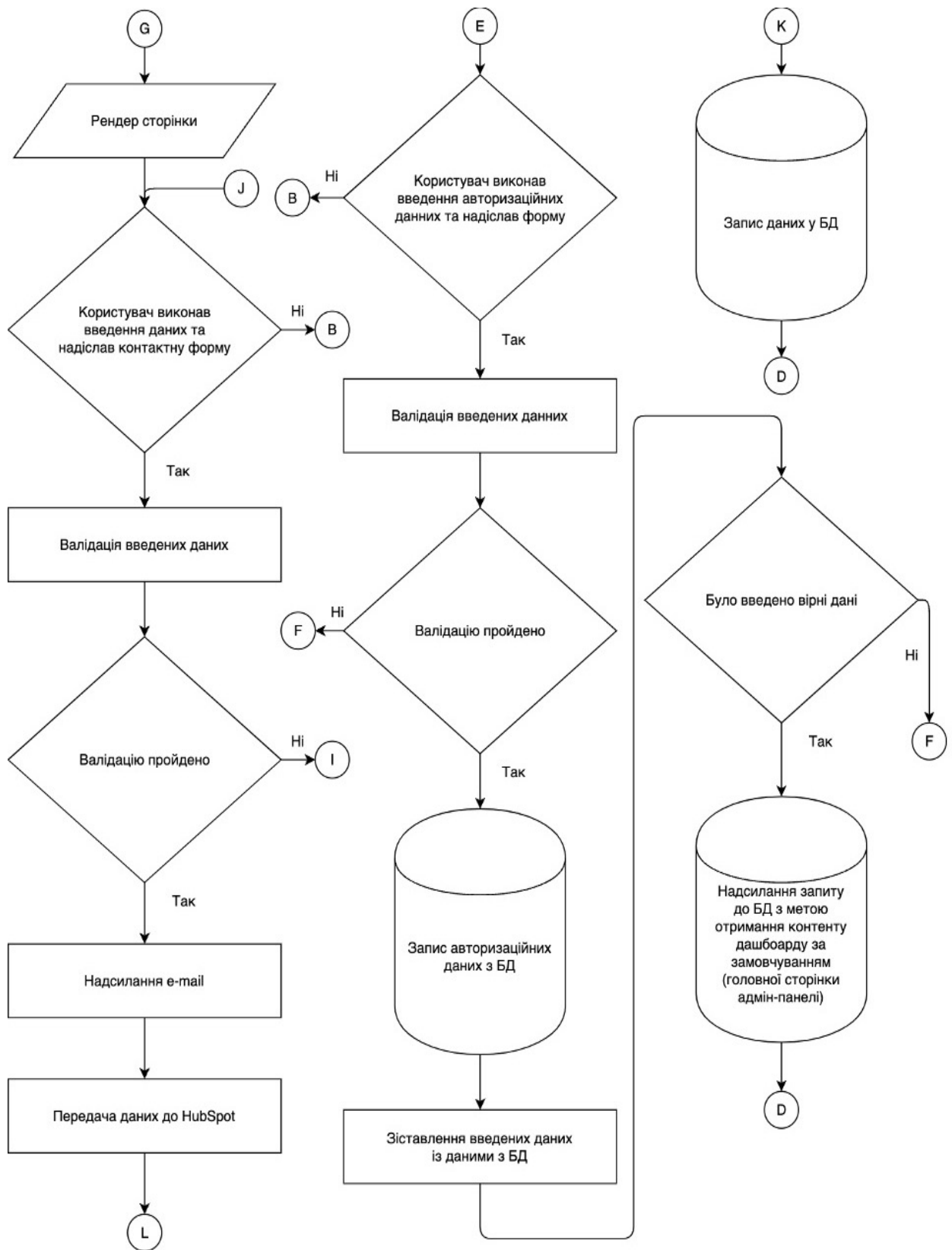
Рис. 2.8. Приклад коду експортованої таблиці БД

2.5. Опис структури програми та алгоритмів її функціонування

У цьому розділі було розглянуто структуру програми та описано основні алгоритми, які забезпечують її функціонування. Програма є веб-сайтом, створеним для людей з обмеженими можливостями, і вона базується на використанні різних технологій, мов програмування та плагінів WordPress. Далі

наведено рисунок 2.9 загальної блок схеми структури програми та основних алгоритмів, що використовуються в роботі.





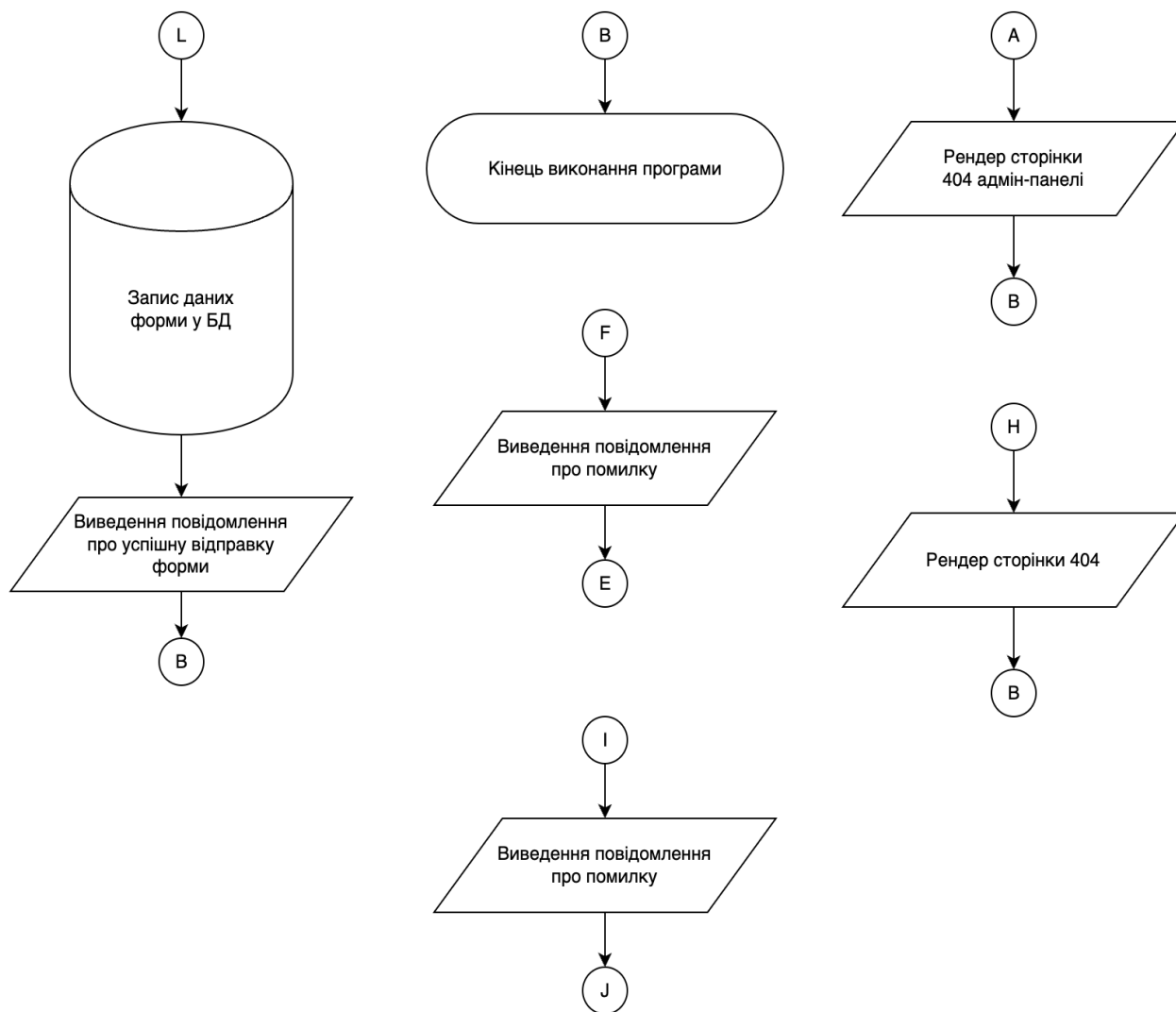


Рис 2.9. Блок-схема роботи продукту

Нижче описано структуру продукту, яка поділяється на такі частини як: фронтенд, бекенд та WordPress.

Фронтенд (клієнтська частина): Реалізована з використанням HTML, CSS та JavaScript, вона відповідає за візуальне представлення веб-сторінок, інтерактивні елементи та взаємодію з користувачем через зручний інтерфейс та можливість отримувати інформацію людям з обмеженими функціональними, когнітивними та фінансовими можливостями .

Бекенд (серверна частина): Основний функціонал реалізований за допомогою мов програмування PHP, що працює в парі з MySQL[18] базою даних. Бекенд відповідає за обробку запитів, доступ до бази даних, обробку та збереження вхідних даних, генерацію вихідних даних та інші серверні операції.

WordPress: Сайт базується на платформі WordPress, що надає готові рішення для управління веб-сайтом. Вона забезпечує роботу з плагінами, налаштування теми оформлення та інші функції для зручного управління контентом та функціональністю сайту.

Також сайт підтримує основні алгоритми функціонування. Як наприклад, алгоритм обробки вхідних даних включає перевірку, валідацію та збереження даних, які вводяться користувачами на веб-сторінках. Це включає в перевірку на коректність заповнення форм, перевірку правильності введених даних та їх збереження в базі даних.

Алгоритм обробки вихідних даних включає обробку запитів користувачів, взаємодію з базою даних та формування необхідної інформації для відображення на веб-сторінках. Це включає в себе вибірку даних з бази даних, сортування, фільтрацію та форматування даних перед їх відправленням на фронтенд.

Алгоритм управління контентом дозволяє додавати, редагувати та видаляти інформацію на веб-сторінках. Це включає в себе створення нових записів, оновлення існуючих, роботу з категоріями, тегами та іншими метаданими. Управління контентом зазвичай реалізується за допомогою плагінів WordPress, таких як Advanced Custom Fields (ACF) та Custom Post Type UI[19].

Для поліпшення швидкості завантаження сторінок та продуктивності програми, можуть використовуватись різні алгоритми оптимізації, наприклад, мініфікація стилів та JavaScript, кешування, стиснення зображень, використання CDN та інші техніки. Це допомагає забезпечити покращену взаємодію з веб-сайтом та підвищення його продуктивності.

У цьому розділі ми розглянули загальну структуру програми та основні алгоритми, які забезпечують її функціонування. Реалізація цих алгоритмів залежить від конкретних вимог та функціональності нашого веб-сайту для людей з обмеженими можливостями.

2.6. Обґрунтування та організація вхідних та вихідних даних програми

Ефективна обробка та управління даними є важливою складовою успішного функціонування сайту. Нижче представлені ключові аспекти, які слід враховувати:

1. Вхідні дані:

Вхідні дані для програми можуть включати різноманітну інформацію, залежно від функціональності та цілей веб-сайту. Наприклад, це можуть бути дані, введені користувачем через форми, такі як ім'я, контактні дані, вибір параметрів тощо. Важливо визначити типи даних, їх структуру та формат, а також методи збору та перевірки вхідних даних для забезпечення їхньої коректності та безпеки.

2. Збереження та управління даними:

Для збереження та управління вхідними та вихідними даними можна використовувати різні підходи. Наприклад, створити базу даних з використанням MySQL, де будуть зберігатись дані про користувачів, вміст сторінок, налаштування та інші важливі дані. Також важливо визначити методи доступу до цих даних, зокрема через використання мови програмування PHP та SQL-запитів.

3. Вихідні дані:

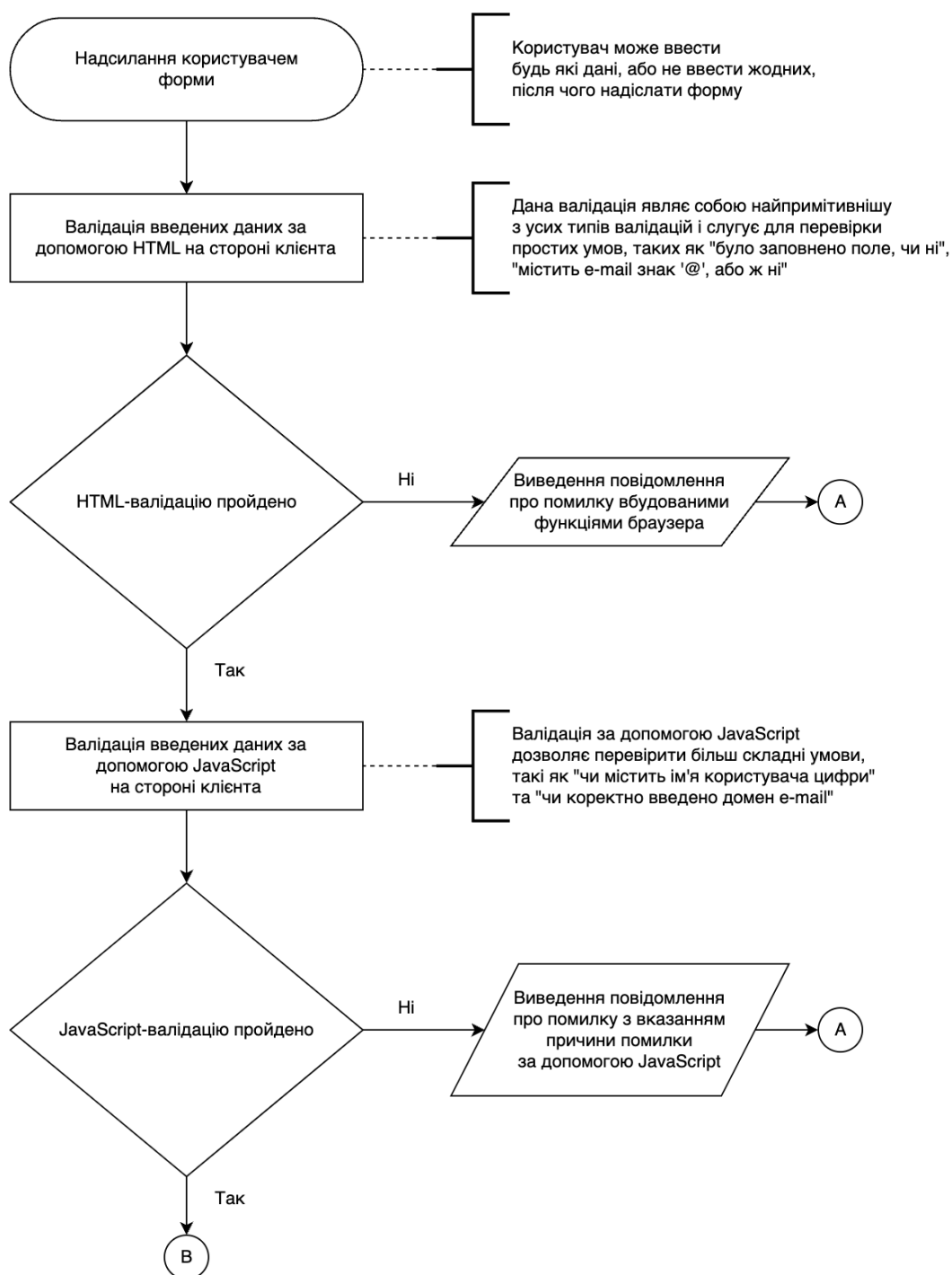
Вихідні дані програми включають ту інформацію, яка надається користувачеві чи використовується для відображення на веб-сторінках. Це можуть бути, наприклад, інтерактивні елементи, результати обробки даних, повідомлення, зображення та інше. Важливо визначити, які дані повинні бути виведені, їхній формат та методи представлення.

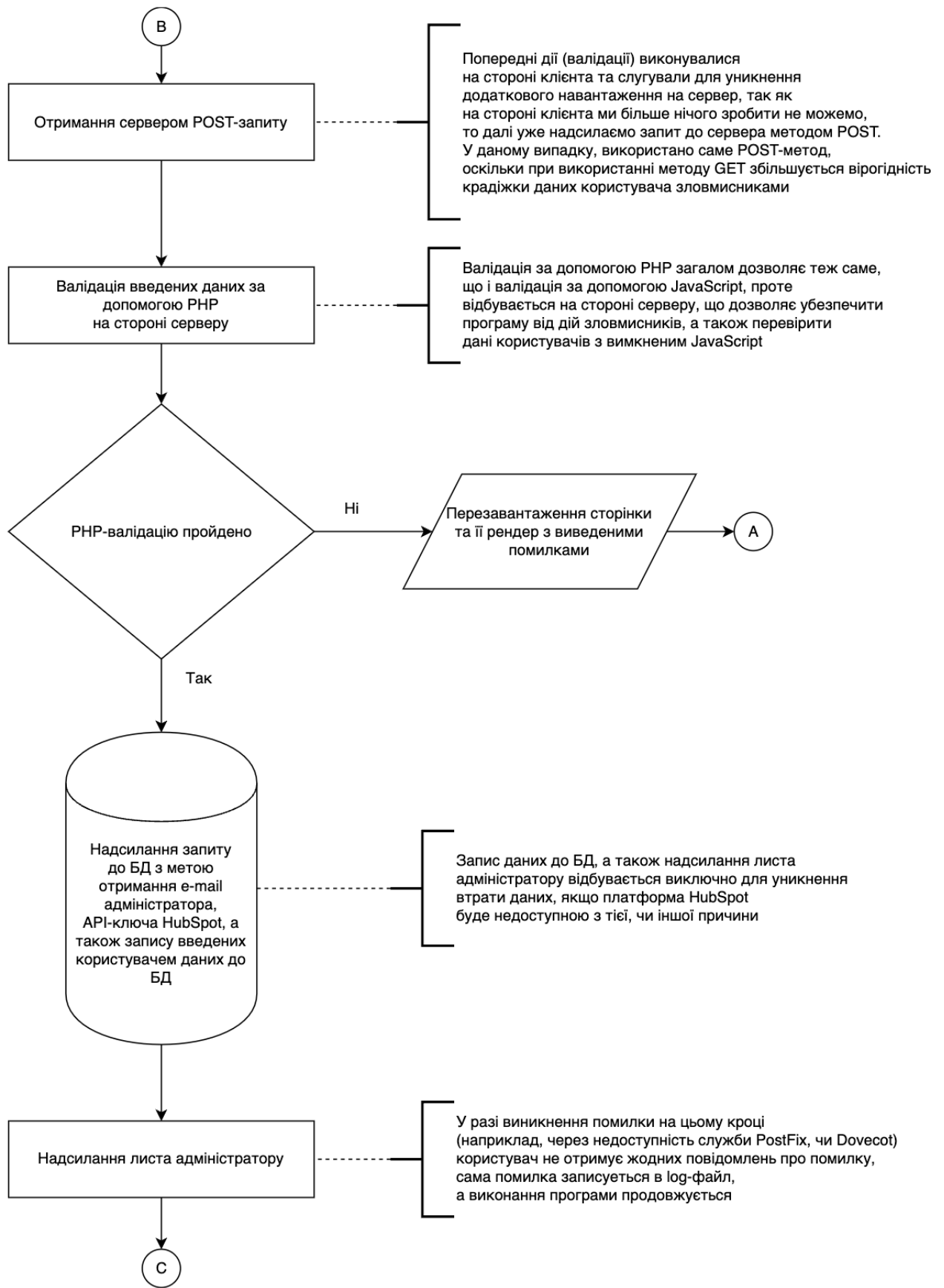
4. Захист та безпека даних:

Забезпечення захисту та безпеки даних є критичним аспектом розробки програми. Ми повинні приділити увагу механізмам шифрування, аутентифікації користувачів, захисту від SQL-ін'єкцій, злому та інших

потенційних загроз безпеці. Розробка відповідних заходів безпеки та політик доступу до даних допоможе забезпечити конфіденційність та цілісність інформації.

Більш детально організацію вхідних та вихідних даних розглянемо на рисунку 2.10, де схематично зображено роботу даних.





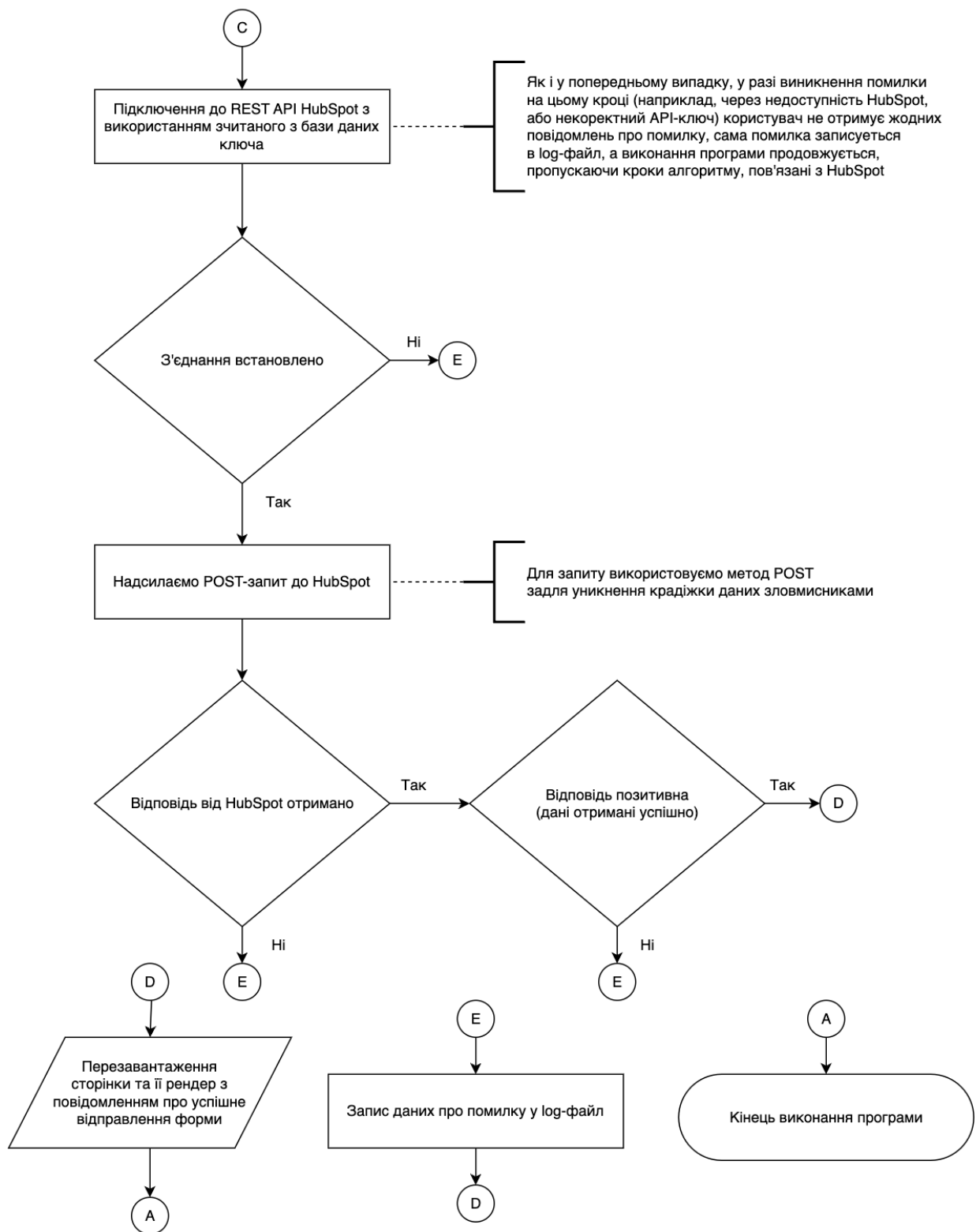


Рис. 2.10. Блок-схема роботи вхідних і вихідних даних розробленого продукту

В цьому розділі було обґрунтували вхідні та вихідні дані програми для веб-сайту, спрямованого на людей з обмеженими можливостями. Також були

визначені методи збирання, збереження, управління та захисту цих даних. Враховуючи ці аспекти, ми забезпечимо ефективну роботу програми та надамо користувачам зручний та безпечний досвід взаємодії з веб-сайтом[20].

2.7. Опис розробленого програмного продукту

2.7.1. Використані технічні засоби

Розроблюваний веб продукт застосовується не лише в умовах сервісного центру, а й при проведенні виїзних робіт, коли доступу до потужного обладнання нема. Тому для запуску вистачає мінімальних ресурсів комп'ютера.

Для роботи потрібні:

- Персональний комп'ютер або ноутбук, планшет, телефон
- Wi-fi або мобільне з'єднання
- Клавіатура
- Мишка або тачпад

2.7.2. Використані програмні засоби

Спочатку використовуємо віртуальний виділений сервер VPS, потім на нього було встановлена операційна система Debian. На нього було встановлено HTTP сервер Apache. Також було встановлено підтримку мови програмування PHP. Далі у якості reverse проху було встановлено веб-сервер Nginx. А у якості СБУД використовується MySQL. Для підтримки функцій POST було використано Postfix(mail transfer agent) та Dovecot (pop/imap server)

При розробці програми використовувався браузер (зазвичай Chrome, для тестування Safari, FireFox та Opera).

Для тестування доступності використовувались доповнення для webKit браузерів [21]. Такі як WAVE Evaluation Tool [22], Screen Reader [23].

Git Kraken було використано для контролю версій. Програмно розробка розгоралась на операційній системі MacOS.

2.7.3. Виклик та завантаження програми

Для того, щоб завантажити веб сторінку, треба стабільне інтернет з'єднання, яке може бути підключено через мережу WI-FI або через мобільну мережу. Можна використовувати будь-який браузер, так як була реалізована кросбраузерність, в пошуковому рядку вводимо такі назву сторінки, після введення натискаємо «Пошук». Відразу відбудеться рендер головної сторінки сайту (рис. 2.11). Продукт завантажено і готове до використання.

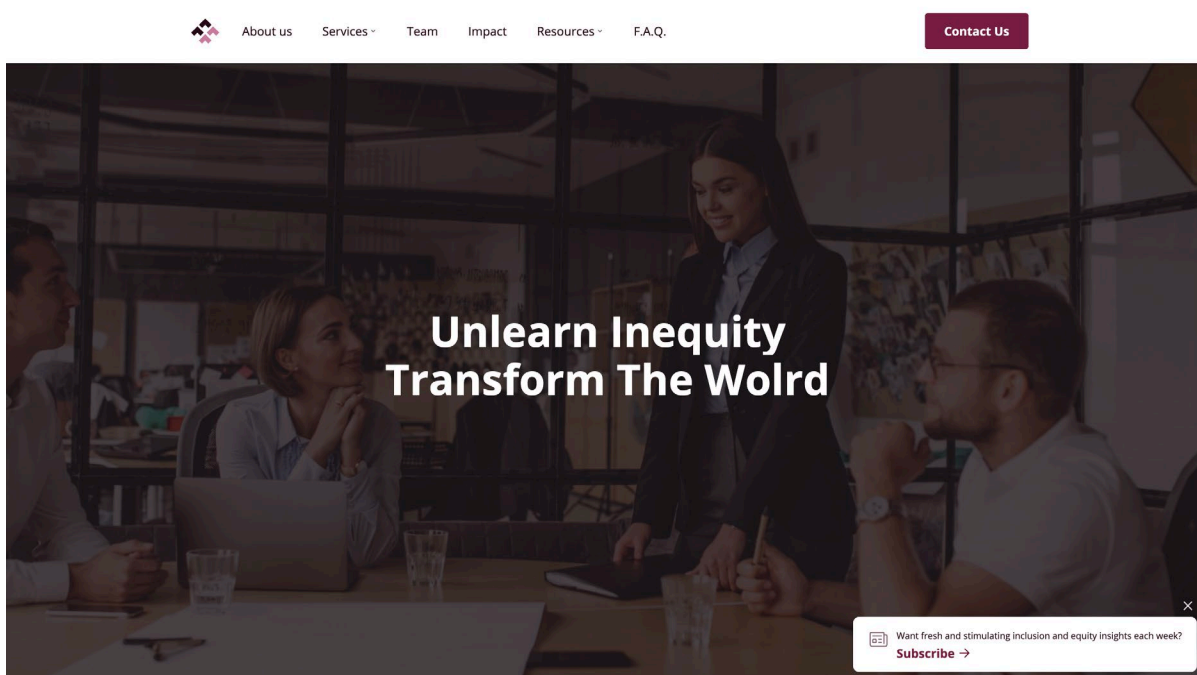


Рис. 2.11. Приклад головної сторінки після відкриття сайту

Для того, щоб зайти в адміністративну панель, адміністратор має ввести в пошуковий рядок посилання на головну сторінку сайту, а також ввести «/wp-admin». Відбувається рендер сторінки та на сторінці відображено форму для авторизації користувача (рис.2.12).

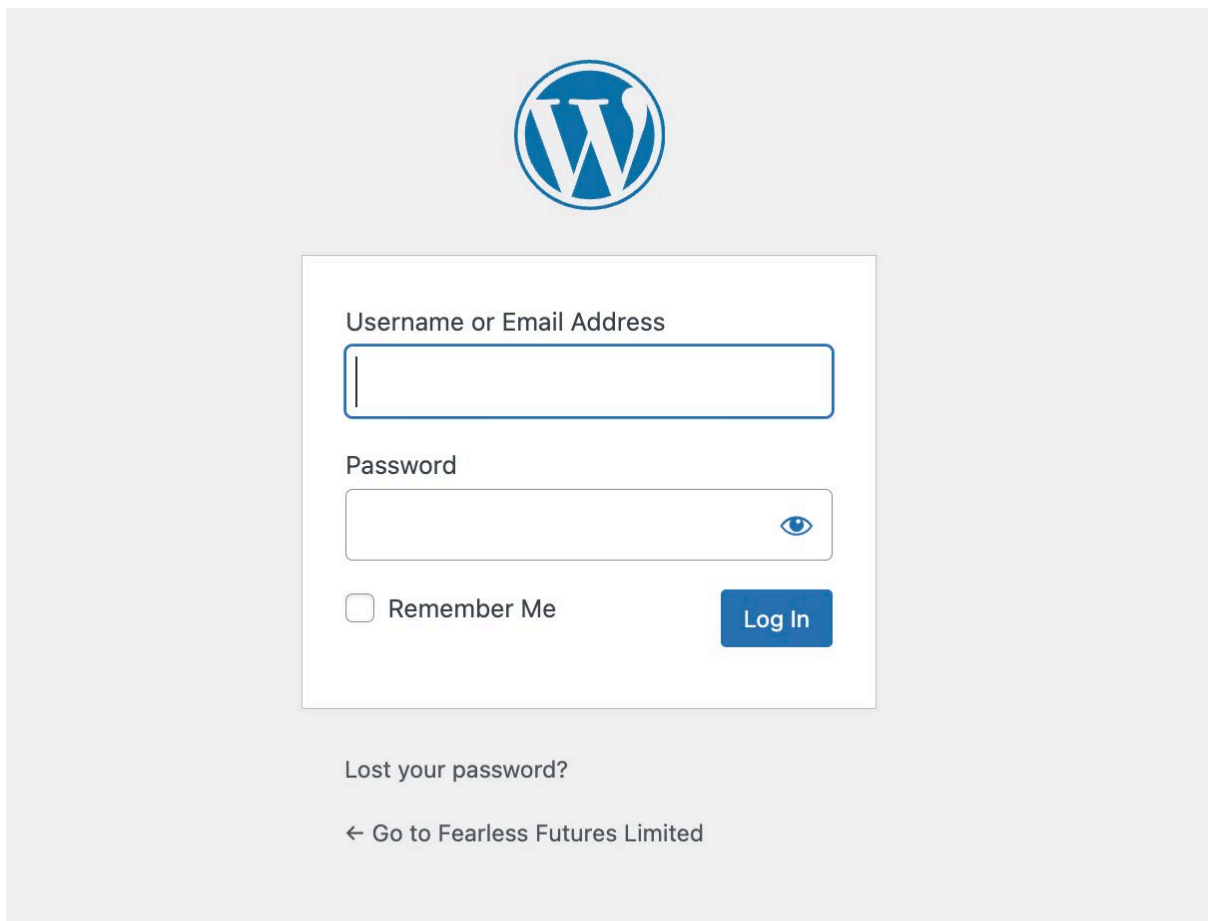


Рис. 2.12 Форма авторизації для адміністратора

На рисунку зображено стандартну форму, яка використовується CVS WordPress для авторизації в адміністративну панель керування контентом.

2.7.4. Опис інтерфейсу користувача

В розробці веб-сайту є важливими такі складові: фронт-енд частина, яка відповідає за візуальну частину, що є важливим для кінцевого продукту, а також, бек-енд частина, без якої неможливе повноцінне існування першої.

Фронт-енд частина в цьому випадку відіграє головну роль, так як сайт було створено за рекомендаціями W3C [24]. Тому звичайна верстка відрізняється наявністю атрибутів ARIA-label. ARIA-label є атрибутом HTML, який використовується для надання текстового опису елементу інтерфейсу, які не мають прикріпленої мітки або тексту. Це дозволяє програмам для читання з

екрану передавати зрозумілу інформацію про ці елементи користувачам з візуальними обмеженнями. Це засіб для забезпечення більшого розуміння та використання інтерфейсу для всіх користувачів.

Опис інтерфейсу було проведено з боку користувача або відвідувача сайту, так і з боку адміністратора сайту. Для більш зрозумілого та чіткого викладення інформації, опис буде відбуватись по блокам. Таким чином, можемо побачити як інтерфейс блоку для користувача так для адміністратора.

На сайті для всіх сторінок є однаковими хедер (рис. 2.13) та футер сайту (рис. 2.14).



Рис. 2.13. Приклад хедеру на реалізованому продукті

Хедер або шапка сайту реалізує зручність навігації по сайту. Хедер є фіксований, тобто коли ми гортаємо сайт вниз навігація є завжди доступною, що полегшує навігацію по сайту. Логотип компанії було розміщено зліва, який є якорем наверх сторінки, якщо ми знаходимось на головній сторінці сайту. Логотип веде на головну сторінку, якщо користувач знаходиться на інших сторінках. Ця функція є зручною, коли користувач дуже часто переходить по сторінках. Такий спосіб полегшує користування сайтом.

Одним із важливих компонентів сайту є зручний зв'язок. Тому важливо було розмістити кнопку «Contact us» на зручне та видиме місце, а саме на фіксованому хедері.

Футер або підвал сайту (рис. 2.14) є незмінним та включає в себе навігацію по сайту, а також соціальні мережі та пошту, за допомогою яких, можна легко та зручно зв'язатись власником сайту.

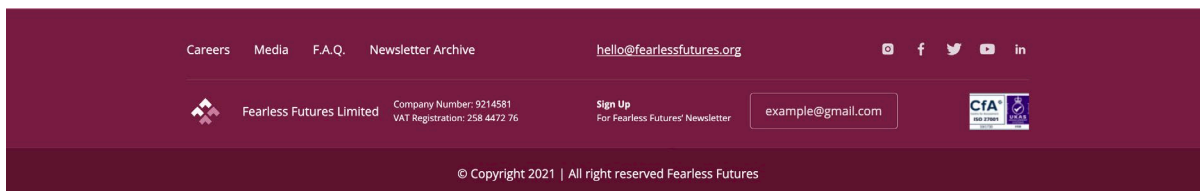


Рис. 2.14. Приклад футера сайту розробленого продукту

Так як наведено на рисунку 2.14, бачить будь-який користувач сайту. Також вище описувалось, що підвал сайту є повторюваним для кожної сторінки. В адмін панелі їх було винесено окремо (рис. 2.15), для економії часу, зусиль та можливих помилок, при повторному заповненні даних.

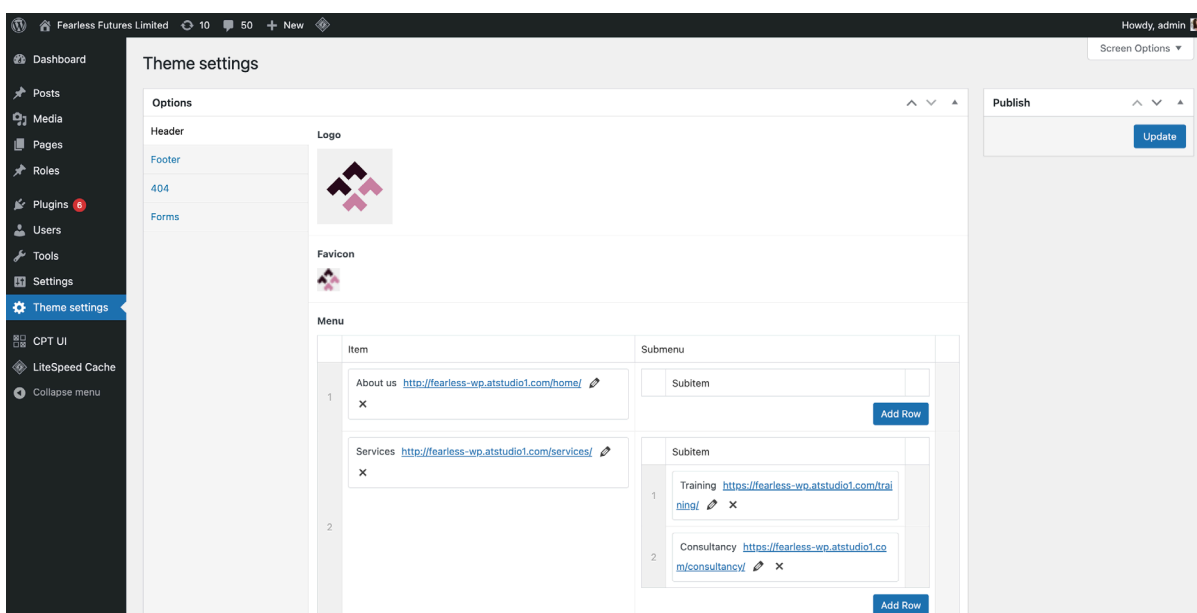


Рис. 2.15. Приклад винесеного хедера та футера до налаштувань теми

В налаштуваннях теми виносяться все, що є повторюваним: хедер, футер, контактна форма, а також така сторінка як 404.

Одним із складних прикладів реалізації aria-label є список з дочірніми елементами. При розгляді сайтів в Розділі 1 наведено приклад, як цю частину коду не враховано, тому проаналізувавши помилку розробника, її було не допущено в кваліфікаційній роботі. На рисунку 2.16 наведено приклад реалізації цієї проблеми. Фокус зчитування з екрану проходить через всі елементи списку.



Рис. 2.16. Приклад реалізації роботи програми зчитування з екрану на дочірніх списках елементу

Розглянемо більше детально реалізацію коду в цій частині сайту (рис. 2.17). В цьому випадку використовувались атрибути для збільшення можливостей охопити різні категорії людей. Наприклад,

- `role="menuitem"` – атрибут використовується в доступності для позначення елементів меню або списків, які представляють пункти меню в інтерфейсі користувача. Цей атрибут допомагає програмам для читання екрану інтерпретувати ці елементи як пункти меню та забезпечує відповідні функції та поведінку при взаємодії з ними.

З використанням `role="menuitem"` програми для читання екрану можуть інтерпретувати ці елементи як пункти меню, надаючи користувачам відповідні можливості навігації та взаємодії, такі як використання клавіш-стрілок, клавіші "Enter" або "Space" для вибору пунктів меню. Це поліпшує доступність інтерфейсу для людей з обмеженими можливостями.

- `aria-haspopup="true"` – атрибут використовується в доступності для позначення елементів інтерфейсу, які мають випадаюче меню або підменю. Цей атрибут допомагає відобразити, що певний елемент має

взаємодію або відношення до показу або відкриття додаткового вмісту або опцій.

- `aria-expanded="false"` - атрибут використовується в доступності для позначення, що елемент, який має розкритий або згорнений вміст, на даний момент згорнутий або невидимий. Цей атрибут допомагає передати стан розкриття або згортання елемента для користувачів з обмеженими можливостями, які використовують технології допомоги, такі як програми для читання екрану.

```
60 <?php if ( have_rows( 'header_menu', 'option' ) ) : ?>
61 <ul class="header_nav-list" role="menu">
62 <?php while ( have_rows( 'header_menu', 'option' ) ) : the_row(); ?>
63 <?php $item = get_sub_field( 'item' ); ?>
64 <?php if ( $item ) : ?>
65 <?php if ( have_rows( 'submenu' ) ) : ?>
66 <li class="header_nav-item header_nav-item-drop" role="menuitem" aria-haspopup="true" aria-expanded="false" tabindex="0"><span class="header_nav-link"><?php echo esc_html( $item['title'] ); ?></span>
67 <ul class="header_drop header_drop-width header_drop-js">
68 <?php while ( have_rows( 'submenu' ) ) : the_row(); ?>
69 <?php $subitem = get_sub_field( 'subitem' ); ?>
70 <?php if ( $subitem ) : ?>
71 <li class="header_drop-item"><a class="header_nav-link" href="<?php echo esc_url( $subitem['url'] ); ?>" target="<?php echo esc_attr( $subitem['target'] ); ?>"><?php echo esc_html( $subitem['ti
72 <?php endif; ?>
73 </ul>
74 </li>
75 <?php else : ?>
76 <li class="header_nav-item" role="menuitem"><a class="header_nav-link" href="<?php echo esc_url( $item['url'] ); ?>" target="<?php echo esc_attr( $item['target'] ); ?>"><?php echo esc_html( $item['title'
77 <?php endif; ?>
78 </ul>
79 </li>
80 </ul>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 <div class="header_search">
87 </div>
88 <?php $header_button = get_field( 'header_button', 'option' ); ?>
89 <?php if ( $header_button ) : ?>
90 <a class="header_btn btn mobile-hide" href="<?php echo esc_url( $header_button['url'] ); ?>" target="<?php echo esc_attr( $header_button['target'] ); ?>"><?php echo esc_html( $header_button['title'] ); ?></a>
91 </div>
92 <button class="header_burger desktop-hide" aria-label="Main Menu" aria-expanded="false">
93 <div></div>
94 </div>
95 </button>
96 </div>
97 </div>
98 </div>
99 <div id="scrollbar">
100 <main id="main">
```

Рис. 2.17. Реалізація випадаючого списку для людей, які користуються зчитуванням з екрана

Розглянемо більш детально адміністративну панель курування контентом в цьому блоці (рис. 2.18). Адміністративна панель має опції зі списком посилань, в яких можна редагувати назву посилання, саме фактичне посилання, місце розташування, а також, наявність дочірніх елементів списку, що дозволяє змінювати та підлаштовувати контент під події та бажання клієнта.

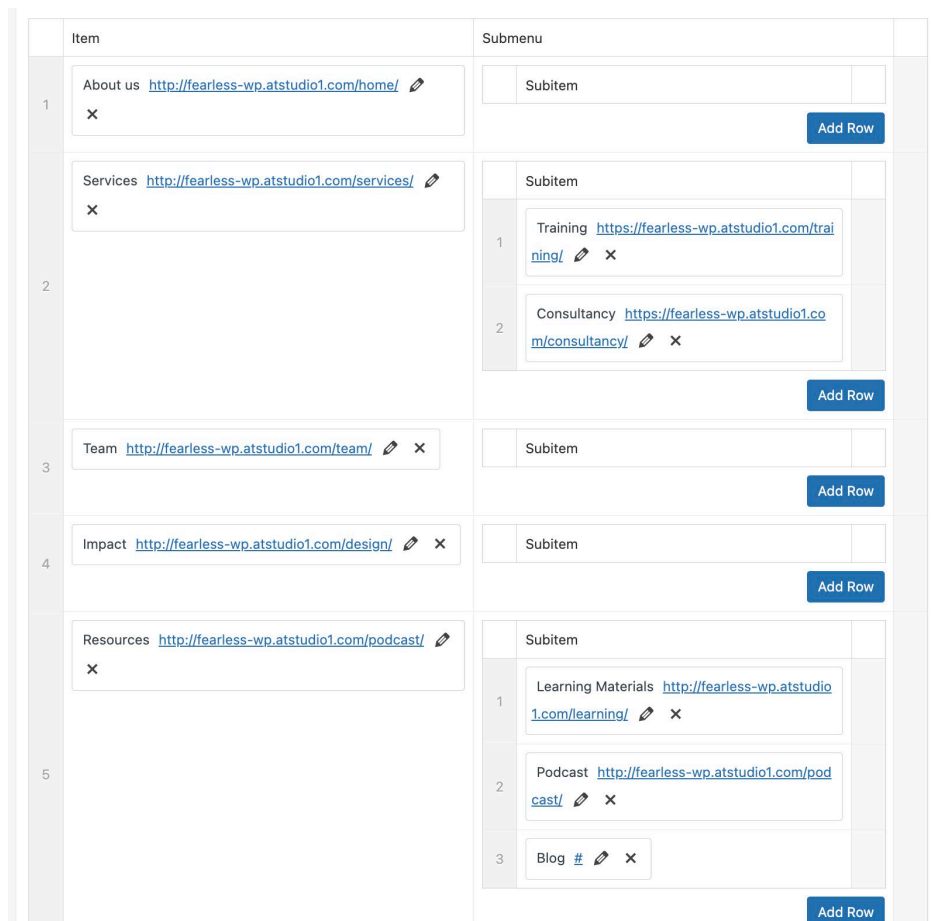


Рис. 2.18. Адміністративна панель з частиною, яка відповідає за списки в хедері сторінки

Блок з відгуками в слайдері є важливим елементом веб-сайту, який відображає відгуки користувачів. Цей блок може мати деякі особливості, що поліпшують доступність та використання для користувачів з обмеженими можливостями. Важливим аспектом доступності є забезпечення відтворення фокусування для елементів слайдера з відгуками. Це означає, що при навігації користувачем за допомогою клавіатури, елементи слайдера повинні отримувати фокус (рис.2.19), щоб користувач міг переглядати та взаємодіяти з вмістом.

What Our Clients Say

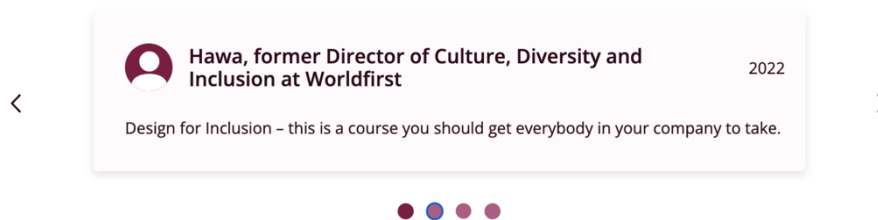


Рис. 2.19. Елементи слайдера, які мають фокус, який допомагає переглядати контент та взаємодіяти з текстом

Доступність через зчитування екрана та фокусування на активних елементах при використанні навігації клавіатурою було досягнуто за допомогою відповідного використання HTML-атрибутів та правильного управління подіями клавіатури для перемикання між відгуками (рис. 2.20).

```
1 <section class="reviews at-block" role="region" aria-roledescription="carousel" aria-label="Our Clients Reviews">
2 <div class="center-wrap">
3 <h2 class="reviews_simple-title simple-title at-element" data-at-name="atTitle"><?php the_field( 'reviews_title' ); ?></h2>
4
5 <?php if ( have_rows( 'reviews_list' ) ) : ?>
6 <div class="reviews-slider">
7 <div class="reviews-slider_wrap">
8 <div class="swiper-container">
9 <div class="swiper-wrapper">
10
11 <?php while ( have_rows( 'reviews_list' ) ) : the_row(); ?>
12 <div class="swiper-slide" aria-hidden="true" role="group" aria-roledescription="slide" aria-label="1 of 5">
13 <div class="reviews-slider_item">
14 <div class="reviews-slider_top">
15 <div class="reviews-slider_photo img-wrap">
16
17 <div>
18 <?php if ( get_sub_field( 'photo' ) ) : ?>
19 
20 <?php endif ?>
21 </div>
22 </div>
23 <div class="reviews-slider_name"><?php the_sub_field( 'name' ); ?></div>
24 <div class="reviews-slider_date"><?php the_sub_field( 'date' ); ?></div>
25 </div>
26 <div class="reviews-slider_content content">
27 <?php the_sub_field( 'text' ); ?>
28 </div>
29 </div>
30 </div>
31 <?php endwhile; ?>
32 </div>
33 <div class="swiper-button-prev mobile--hide">
34 <svg viewBox="0 0 24 24" fill="none" xmlns="http://www.w3.org/2000/svg">
35 <path d="M16 4L8 12L16 20" stroke-width="2" stroke-linecap="round" stroke-linejoin="round"></path>
36 </svg>
37 </div>
38 <div class="swiper-button-next mobile--hide">
39 <svg viewBox="0 0 24 24" fill="none" xmlns="http://www.w3.org/2000/svg">
40 <path d="M8 4L16 12L8 20" stroke-width="2" stroke-linecap="round" stroke-linejoin="round"></path>
41 </svg>
42 </div>
43 </div>
44 <div class="swiper-pagination" role="group" aria-label="Choose slides"></div>
45 </div>
46 <?php endif; ?>
47
```

Рис. 2.20. Приклад частинки коду, яка реалізує доступність на каруселі з відгуками

Атрибути `aria-hidden`, `role`, `aria-roledescription` та `aria-label` використовуються для поліпшення доступності веб-сайту, зокрема для взаємодії зі слайдером або каруселлю, де показується певна кількість елементів у вигляді послідовних слайдів.

1. `aria-hidden="true"`: Цей атрибут вказує на те, що елемент знаходиться в стані приховання і його не слід ураховувати під час навігації або взаємодії. В контексті слайдера, коли елементи, які не є поточним слайдом, мають атрибут `aria-hidden="true"`, технології допомоги можуть ігнорувати їх, забезпечуючи фокусування лише на видимому слайді.

2. `role="group"`: Цей атрибут вказує на роль елемента, який містить групу елементів, таких як слайди в каруселі. Роль "group" допомагає створити логічну групування елементів, що дозволяє технологіям допомоги правильно інтерпретувати структуру слайдера.

3. `aria-roledescription="slide"`: Цей атрибут надає опис ролі елемента. У даному випадку, опис "slide" вказує на те, що елемент є слайдом у каруселі. Це допомагає користувачам з обмеженими можливостями зрозуміти, що цей елемент є складовою частиною каруселі та має функцію слайду.

4. `aria-label="1 of 5"`: Цей атрибут надає текстовий опис елемента для користувачів з обмеженими можливостями. У даному випадку, "1 of 5" вказує на поточну позицію слайда в каруселі. Ця інформація допомагає користувачам зрозуміти, скільки слайдів є в каруселі та який саме слайд знаходиться на даний момент у вигляді видимого контенту.

Приклад використання WordPress при використанні слайдера. Блок з слайдером має назву, має частину для фотографії, ім'я людини, яка дала відгук, дата вугука та безпосередньо текст з інформацією. Кількість відгуків можна змінювати, а також, якщо немає фотографії, то залишається за змовчуванням, така як зараз зображено на рисунку 2.21.

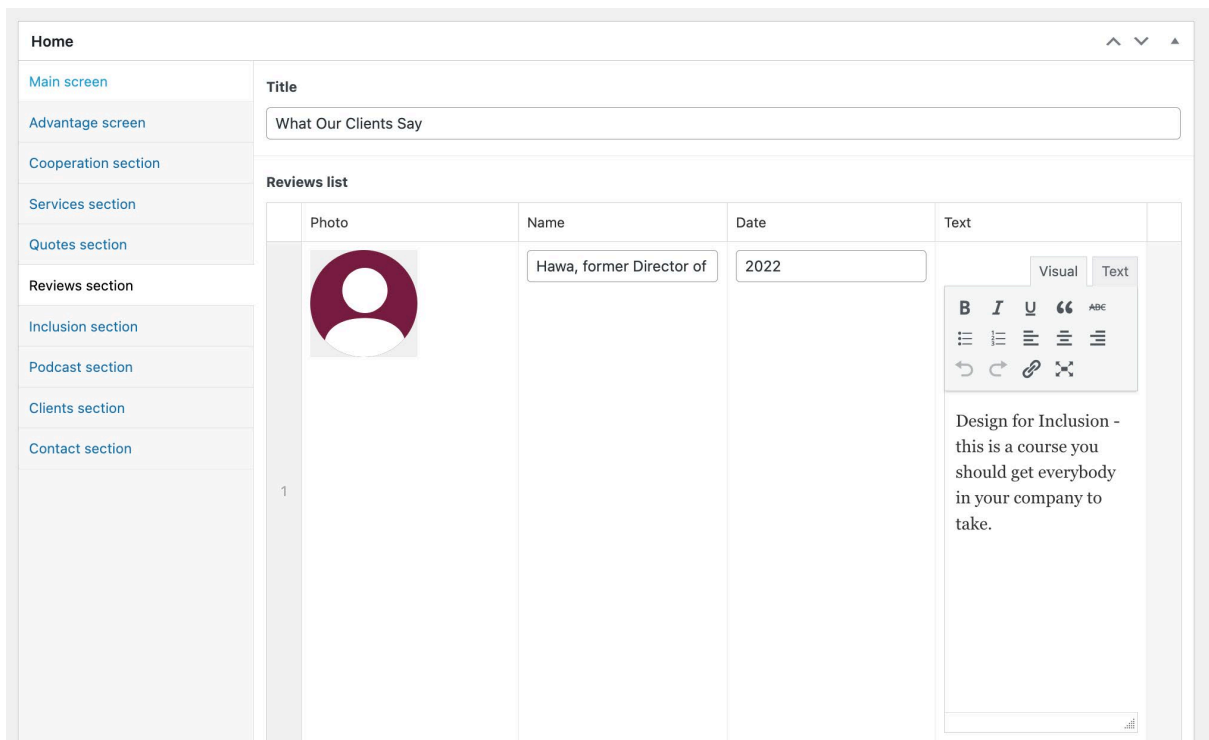


Рис. 2.21. Слайдер в адмін панелі WordPress

Далі розглянемо роботу форму відправки повідомлень від користувача до адміністратора (рис. 2.22.).

Complete the form
to get in touch with us!

Your name

Company name

E-mail

Message

I agree to company [terms and conditions.](#)

Receive a consultation

Рис. 2.22. Приклад форми відправки повідомлень

Розберемось більш детально частину цього блоку. Форма відправки повідомлень, що містить поля для імені, назви компанії, електронної пошти та повідомлення, виконує важливу функцію збору даних від користувачів та надсилання їх на вказану адресу. Описуючи роботу такої форми, можна виокремити наступні етапи:

1. Форма може бути розміщена на веб-сторінці з використанням HTML-тегів, таких як `<form>`, `<input>`, `<textarea>`, а також додаткових елементів. Більше детально можна розглянути на зображенні лістингу коду (рис. 2.23). А також приклад заповнення форми приведено нижче (рис. 2.24)

Complete the form to get in touch with us!

Your name

Company name

E-mail

Message

I agree to company [terms and conditions.](#)

Рис. 2.23. Приклад заповнення поля форми

```

21 <div class="contact_right">
22 <?php if ( have_rows( 'contact_form' ) ) : ?>
23 <?php while ( have_rows( 'contact_form' ) ) : the_row(); ?>
24 <form id="form_contacts" class="inputs" action="">
25 <fieldset>
26 <legend class="contact_right-title">?php the_sub_field( 'title' ); ?<br> <?php the_sub_field( 'title_second' ); ?</legend>
27 <div class="inputs_field-wrap">
28 <input class="inputs_field" type="text" id="text" aria-required="true" name="firstname" aria-describedby="text-error" autocomplete="name" required="" minlength="2">
29 <label class="inputs_content" for="text"><?php the_sub_field( 'name_field' ); ?</label>
30 <label class="inputs_field-error" id="text-error" for="">The text field is mandatory.</label>
31 </div>
32 <div class="inputs_field-wrap">
33 <input class="inputs_field" type="text" id="textCompany" name="company" aria-describedby="company-error" name="text" aria-required="true" autocomplete="organization" required="" minlength="2">
34 <label class="inputs_content" for="textCompany"><?php the_sub_field( 'company_field' ); ?</label>
35 </div>
36 <div class="inputs_field-wrap">
37 <input class="inputs_field" type="email" id="email" name="email" aria-describedby="mail-error" aria-required="true" autocomplete="email" required="">
38 <label class="inputs_content" for="email"><?php the_sub_field( 'email_field' ); ?</label>
39 <label class="inputs_field-error" id="mail-error">The email field is mandatory.</label>
40 </div>
41 <div class="inputs_field-wrap inputs_field-wrap-textarea">
42 <textarea class="inputs_field" name="message" id="textarea" aria-required="true" required="">
43 <label class="inputs_content" for="textarea"><?php the_sub_field( 'message_field' ); ?</label>
44 </div>
45 <div class="inputs_checkbox">
46 <label class="inputs_checkbox-label">
47 <input class="inputs_checkbox-input" type="checkbox" checked="" required=""><span class="inputs_checkbox-content" role="checkbox" aria-labelledby="worldpeace"></span><span class="inputs_checkbox-lab-
48 </div>
49 <?php $terms_link = get_sub_field( 'terms_link' ); ?>
50 <?php if ( $terms_link ) : ?>
51 <a class="inputs_checkbox-link" href="<?php echo esc_url( $terms_link['url'] ); ?>" target="<?php echo esc_attr( $terms_link['target'] ); ?>"><?php echo esc_html( $terms_link['title'] ); ?</a>
52 <?php endif; ?>
53 </label>
54 </div>
55 <input type="hidden" name="to" value="<?php the_field( 'forms_admin_email', 'option' ); ?>" />
56 <div class="inputs_btn">
57 <button class="btn" type="submit"><?php the_sub_field( 'button' ); ?</button>
58 </div>
59 </fieldset>
60 </form>

```

Рис. 2.24. Приклад коду форми

2. На боці клієнта (за допомогою JavaScript) та на боці сервера (за допомогою PHP) виконується валідація введених даних користувачем. Валідація допомагає перевірити, чи коректно заповнені всі обов'язкові поля форми та чи відповідають вони вимогам (наприклад, правильний формат

електронної пошти). У разі виявлення помилок, користувач отримує повідомлення про неправильне заповнення полів.

3. Після успішної валідації введених даних, дані форми будуть надіслані на сервер за допомогою HTTP-запиту. Це виконується за допомогою методу POST. На сервері збираються дані з форми та обробляються відповідним способом, а саме зберігаються у базі даних або надсилаються на вказану електронну адресу.

4. Після успішного надсилання повідомлення користувачеві буде показане підтвердження, що повідомлення було успішно надіслане. На даному проекті це реалізовано як спливаюче вікно з подякою (рис. 2.25). Також, зміст модульного вікна можна змінювати в темі сайту в адмін-панелі WordPress (рис. 2.26).

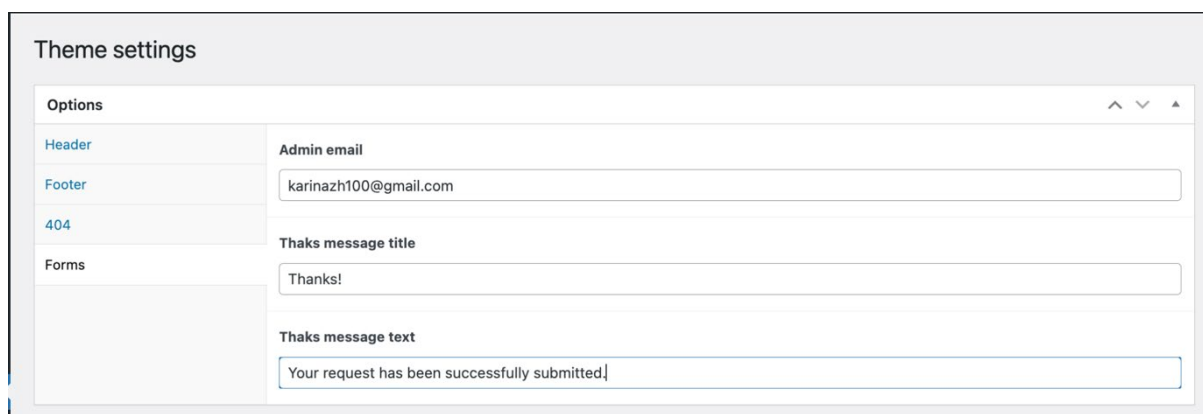


Рис. 2.25. Приклад налаштувань модального вікна в адмін-панелі

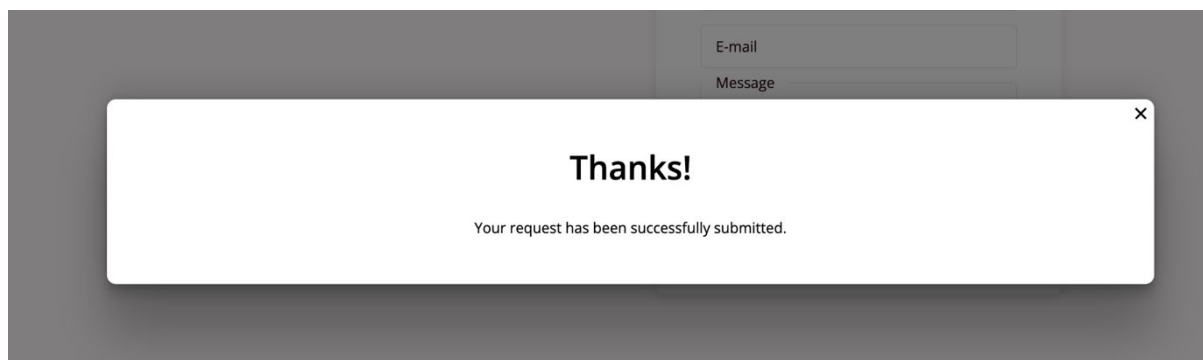


Рис. 2.26. Спливаюче вікно з підтвердження успішної відправки

Одним із важливих моментів в створенні сайту - це додавання блогу (приклад на рис. 2.27).

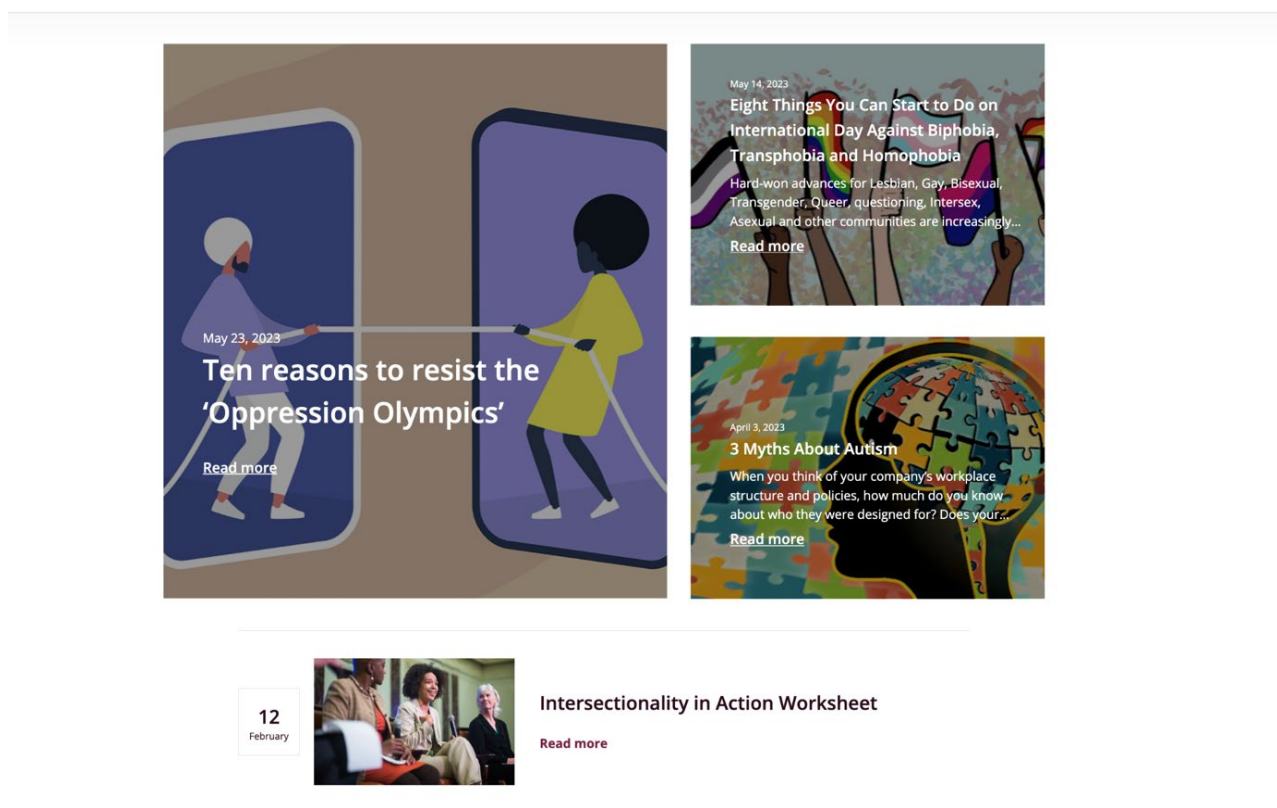


Рис. 2.27. Приклад блогу на сторінці

Дуже важливим є зручним наповнення контентом, що особливо стосується блогу. Так як, це буде відбуватись періодично, і часто, різними людьми, важливо робити інтефейс зручним та зрозумілим кожному, навіть, для користувачів, які не “дружать” з комп’ютером. Тому було розроблено максимально простий та зрозумілий інтерфейс (рис. 2.28).

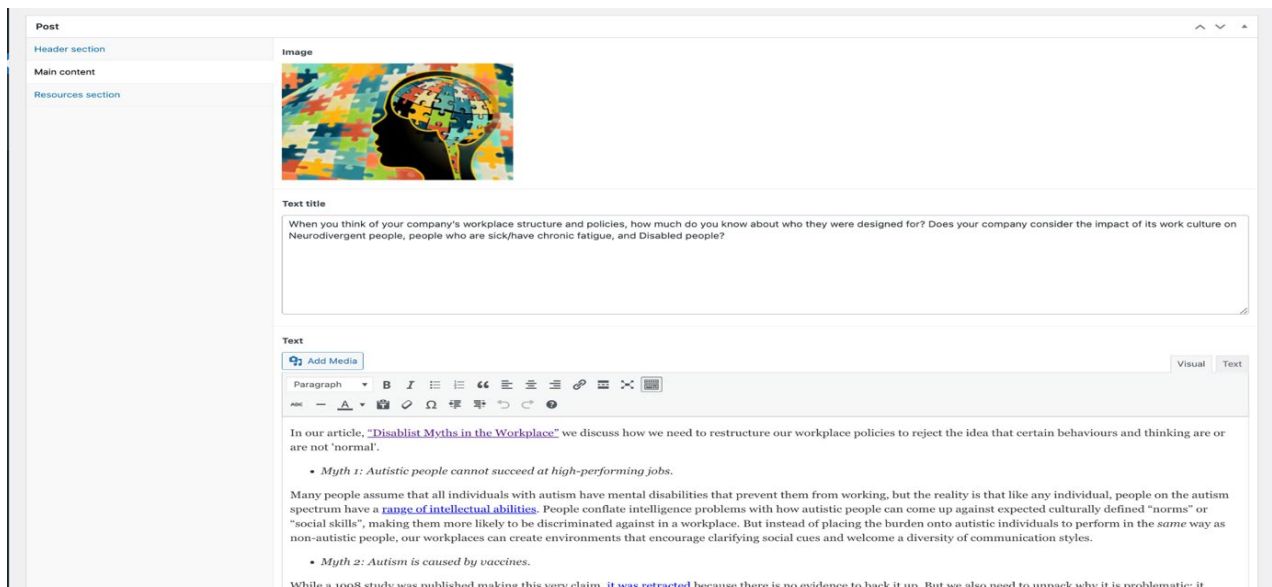


Рис. 2.28. Приклад зручного та зрозумілого інтерфейсу в адмін палені на сторінці створення постів для блогу.

Блог розбито на три головні частини: хедер та головний банер (тобто початок сторінки), пост та рекламу банер з футером. Більш детально розглянемо структуру посту. Вона складається з фотографії, яка буде на сторінці посту, тексту, який також виводиться на сторінку блогу та основного тексту. Також, обов'язковими є дата та час викладення посту, автор та посилання генерується автоматично після введення назви.

Отже, враховуючи проведений аналіз в розділі один, було зібрано багато різного типів помилок та вдалих рішень при розробці продукту для різних типів доступності. На основі цих даних було проведено порівняння створеного власноруч сайту з деякими відомими сторінками в Інтернеті в Таблиці 2.3, яка наведена нижче. Метою цього порівняння є оцінка рівня розробленості, функціональності та дизайну сайту в порівнянні з іншими веб-ресурсами. Результати порівняння допоможуть виявити сильні та слабкі сторони сайту, а також визначити можливості для подальшого вдосконалення.

Порівняльна таблиця робіт

	Розроблена система	Yasno	ЦУМ	ВВС
Закріплена(фіксована) шапка сайту для легкої навігації по ньому	+	+	-	-
Ефекти при наведенні на клікабельні елементи	+	+	+	+
Велика контрастність між фоном та текстом	+	-	-	+
Заповнений атрибут alt	+	-	-	+/-
Можливість переключання по сайту через таби/клавіатру	+	-	+/-	+
Коректна реалізація пошуку по сайту через клавіатуру	-	-	-	+
Коректне відпрацювання Screen Reader сайту	+	-	-	+

Розглянувши таблицю, можна прийти до висновку, що проведений аналіз та виправлення помилок на основі загальновідомих сайтів дав результати.

РОЗДІЛ 3

ЕКОНОМІЧНА ЧАСТИНА

3.1. Визначення трудомісткості та вартості розробки програмного продукту

Початкові дані:

1. передбачуване число операторів програми – 2300
2. коефіцієнт складності програми – 1,45;
3. коефіцієнт корекції програми в ході її розробки – 0,2;
4. годинна заробітна плата програміста– 170 грн/год;

За статистикою на платформі за пошуком роботи «Work.ua» заробітна праця Front-end та WordPress [25] розробників приблизно дорівнює 30 000 грн, а графік розробника буде стандартний – 176год/міс. Тому розрахунки будуть такими: $30000/176 = 170$

6. коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,3;
7. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 1,2;
8. вартість машино-години ЕОМ –0,38 грн/год.

Так як для розробки цього проекту використовувався потужний ноутбук, треба враховувати вартість на енергозатрати та інтернету від провайдера «Тріолан». Типовий ноутбук споживає приблизно коливається від 15Вт/год до 65Ват/год. Виходячи з характеристик ноутбуку, врахуємо, що споживання електроенергії за годину виходить приблизно 40 Вт*год. За даними постачальнику електроенергії Yasno, тариф для населення на початок червня становить 2,64 грн/кВт*год. Отже, вартість ноутбуку становить $0,04 * 2,64 = 0,11$ грн/год. Домашнє інтернет-підключення від «Тріолан» оплачується по тарифу в 195грн/місяць . Отже, година користування становить 195грн /30

днів/24 год = 0,27 грн. Тому загальна вартість машино-години ЕОМ становить $0,11 + 0,27 = 0,38$.

Нормування праці в процесі створення веб продукту істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки веб продукту може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки веб продукту можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{oml} + t_d, \text{ людино-годин, (4.1)}$$

де t_o - витрати праці на підготовку й опис поставленої задачі (приймається 50 людино-годин);

t_u - витрати праці на дослідження алгоритму рішення задачі;

t_a - витрати праці на розробку блок-схеми алгоритму;

t_n - витрати праці на програмування по готовій блок-схемі;

$t_{отл}$ - витрати праці на налагодження програми на ЕОМ;

t_d - витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у програмному забезпеченні, яке розробляється.

Умовне число операторів:

$$Q = q \cdot C \cdot (1 + p), \text{ (4.2)}$$

де q - передбачуване число операторів (2300);

C - коефіцієнт складності програми (1,45);

p - коефіцієнт корекції програми в ході її розробки (0,4).

Звідси умовне число операторів в програмі:

$$Q = 2300 \cdot 1,45 \cdot (1 + 0,4) = 4669$$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = (Q \cdot B) / (75 \dots 85)K, \text{ людино-годин,} \quad (3.3)$$

де B – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі, $B=1.3$;

K – коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності. до 2 – 1,1;

$$t_u = (4669 \cdot 1,3) / (85 \cdot 1,1) = 60, \text{ людино-годин.}$$

Витрати праці на розробку алгоритму рішення задачі:

$$t_a = Q / (20 \dots 25)K, \text{ людино-годин.} \quad (3.4)$$

$$t_a = 4669 / (25 \cdot 1,2) = 155, \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = Q / (20 \dots 25)K, \text{ людино-годин.} \quad (3.5)$$

$$t_n = 4669 / (24 \cdot 1,2) = 177, \text{ людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

- за умови автономного налагодження одного завдання:

$$t_{oml} = Q / (4 \dots 5)K, \text{ людино-годин,} \quad (3.6)$$

$$t_{oml} = 4669 / (5 \cdot 1,2) = 778, \text{ людино-годин,}$$

- за умови комплексного налагодження завдання:

$$t_{омл}^k = 1.5 \cdot t_{омл}, \text{ людино-годин} \quad (3.7)$$

$$t_{отл}^k = 1,5 \cdot 778 = 583,6 \text{ людино-годин}$$

Витрати праці на підготовку документації:

$$t_{\partial} = t_{\partial p} + t_{\partial o} \quad (3.8)$$

де $t_{\partial p}$ – трудомісткість підготовки матеріалів і рукопису

$$t_{\partial p} = Q / (15 \dots 20) K \quad (3.9)$$

$t_{\partial o}$ – трудомісткість редагування, печатки й оформлення документації

$$t_{\partial o} = 0.75 \cdot t_{\partial p} \quad (3.1)$$

Підставляємо дані у формулу і отримуємо:

$$t_{\partial p} = 4669 / (20 \cdot 1,2) = 195, \text{ людино-годин.}$$

$$t_{\partial o} = 0,75 \cdot 194,5 = 146 \text{ людино-годин.}$$

$$t_{\partial} = 195 + 146 = 341 \text{ людино-годин.}$$

Отримаємо трудомісткість розробки програмного забезпечення:

$$t = 50 + 60 + 155 + 177 + 778 + 341 = 1561, \text{ людино-годин.}$$

У результаті ми розрахували, що в загальній складності необхідно 2315,65 людино-годин для розробки даного програмного забезпечення.

3.2 Розрахунок витрат на створення програми

Витрати на створення ПЗ К_{ПО} включають витрати на заробітну плату виконавця програми З_{ЗП} і витрат машинного часу, необхідного на налагодження програми на ЕОМ:

$$K_{ПО} = Z_{ЗП} + Z_{МВ}, \text{ грн} \quad (3.10)$$

Заробітна плата виконавців визначається за формулою:

$$Z_{ЗП} = t \cdot C_{ПР}, \text{ грн} \quad (3.11)$$

де: t - загальна трудомісткість, людино-годин;

$C_{ПР}$ - середня годинна заробітна плата програміста, грн/година

З урахуванням того, що середня годинна зарплата програміста становить 177 грн / год, отримуємо:

$$Z_{ЗП} = 1561 \cdot 170 = 265\,370 \text{ грн.}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ, визначається за формулою:

$$Z_{МВ} = t_{отл} \cdot C_{мч}, \text{ грн}, \quad (3.12)$$

де $t_{отл}$ - трудомісткість налагодження програми на ЕОМ, год;

$C_{мч}$ - вартість машино-години ЕОМ, грн/год (0,8 грн/год).

Підставивши в формулу (3.3) відповідні значення, визначимо вартість необхідного для налагодження машинного часу:

$$Z_{МВ} = 778 \cdot 0,38 = 296 \text{ грн.}$$

Звідси витрати на створення програмного продукту:

$$K_{\text{ПО}} = 265\,370 + 296 = 265\,669 \text{ грн.}$$

Очікуваний період створення ПЗ:

$$T = t / (B_k \cdot F_p) \text{ міс.} \quad (3.13)$$

де B_k - число виконавців (дорівнює 1);

F_p - місячний фонд робочого часу (при 40 годинному робочому тижні $F_p=176$ годин).

Очікуваний період створення ПЗ:

$$T = 1561 / 1 \cdot 176 \approx 8,7 \text{ міс.}$$

Висновок: Створений веб-сайт з доступністю для людей з обмеженими можливостями. Мета цього сайту – надати доступ до інформації різні категорії населення за фізичними, фінансовими та когнітивними навичками. Вартість даного програмного забезпечення становить 265 669 грн. Планується, що процес розробки займе приблизно 8,7 місяців. Цей термін обумовлений невеликим досвідом розробника та кількістю задач покладених на нього.

ВИСНОВОК

Робота над кваліфікаційним проектом, яка включала створення веб-сайту для людей з обмеженими можливостями, виконана в повному обсязі. В процесі розробки було використано широкий спектр технологій, мов програмування та плагінів WordPress, що сприяло створенню доступного та функціонального сайту.

Процес розробки сайту включав детальне описання використаних технологій, таких як HTML, CSS, JavaScript, PHP та MySQL. Було описано їх функціональність та роль у роботі сайту, включаючи створення веб-сторінок, дизайну та розміщення контенту, взаємодії з користувачем та зберігання даних у базі даних.

Обґрунтовано та організовано вхідні та вихідні дані програми, зокрема для форми відправки повідомлень. Описано структуру програми та алгоритми її функціонування, що допомагають зрозуміти, як сайт обробляє дані користувачів та забезпечує їх валідацію, зберігання та надсилання.

Архітектура сайту була розроблена з урахуванням комерційного спрямування, з функціональністю форми відправки повідомлень та блогу. Було описано, як структура сайту допомагає користувачам з обмеженими можливостями отримувати повний доступ до контенту та надає їм можливість комфортно взаємодіяти з сайтом.

При розробці веб-сайту приділено увагу доступності, зокрема застосуванню атрибутів ARIA, які допомагають користувачам з обмеженими можливостями отримати доступ до функціональності сайту та коректно інтерпретувати його структуру.

В кваліфікаційній роботі продемонстровано розуміння використаних технологій, врахування принципів доступності та забезпечено функціональність, яка відповідає потребам цільової аудиторії. Результатом роботи є веб-сайт, який забезпечує зручну взаємодію та доступність для користувачів з обмеженими можливостями.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Головний сайт ТОВ «Ясно». URL: <https://yasno.com> (дата звернення: 15.03.2023)
2. Сервіс для перевірки контрастності між фоном та текстом. URL: <https://coolors.co/contrast-checker/112a46-acc8e5> (дата звернення: 15.03.2023)
3. Альтернативний текст на зображеннях. URL: <https://project-seo.net/uk/terms-uk/alt-2/> (дата звернення: 10.04.2023)
4. Офіційний сайт Центрального Універмагу Києва. URL: <https://about.tsum.ua/en> (дата звернення: 15.03.2023)
5. Сторінка новин з великом охопленням аудиторії. URL: <https://www.bbc.com/> (дата звернення: 15.03.2023)
6. Плагін на WordPress. URL: <https://wordpress.org/plugins/psn-pagespeed-ninja/> (дата звернення: 22.01.2023)
7. Сервіс для перевірки швидкості завантаження сайту. URL: <https://pagespeed.web.dev/> (дата звернення: 22.01.2023)
8. Плагін на WordPress. URL: <https://wordpress.org/plugins/advanced-custom-fields/> (дата звернення: 22.01.2023)
9. Функція Mutch. URL: <https://www.javascripttutorial.net/javascript-string-match/> (дата звернення: 10.02.2023)
10. Getting started with ASP.NET MVC 5. URL: <https://learn.microsoft.com/en-us/aspnet/mvc/overview/getting-started/introduction/getting-started> (дата звернення: 15.02.2023).
11. Онлайн мануал по WordPress. URL: https://codex.wordpress.org/Main_Page (дата звернення: 10.12.2022)
12. Brad Williams, David Damstra та Hal Stern. Professional WordPress: Design and Development - "Wiley", January 12, 2015
13. Elizabeth Castro. HTML for the World Wide Web. Peachpit Press, 2003, с. 22-45

14. David Flanagan. JavaScript. The Definitive Guide, O'Reilly Media, Incorporated, 2011, с.523-585
15. Офіційний сайт бібліотеки jQuery. URL: <https://greensock.com/gsap> (дата звертання: 20.10.22)
16. Raghu Ramakrishnan and Johannes Gehrke. Database Management Systems, McGraw-Hill, 2000, с. 3-57
17. Тарас Ванчук та Вімс Харманс. HTTPS: Everywhere: How to Secure Your Websites and Web Applications, Packt Publishing, 2018 с. 39-45, с.70-77
18. Hugh E. Williams та David Lan. Web Database Applications with PHP and MySQL - O'Reilly Media, 2004, с. 20-40
19. Офіційний плагін WordPress. URL: <https://wordpress.org/plugins/custom-post-type-ui/> (дата звернення: 18.02.23)
20. Бен Фрінклін. Responsive Web Design with HTML5 and CSS3, Packt Publishing, 2012, с. 278-289J
21. Офіційний сайт вебкіт браузерів: URL: <https://webkit.org/> (дата звернення: 11.04.2023)
22. Доповнення до вебкіт браузеру для тестування доступності. URL: <https://chrome.google.com/webstore/detail/wave-evaluation-tool/jbbplnrkjmeebjrijfedlgcdilocofoh> (дата звернення: 11.04.2023)
23. Доповнення до вебкіт браузеру для тестування доступності. URL: <https://chrome.google.com/webstore/detail/screen-reader/kgejglhrpjiefppelplmjglcjbhoiplfn> (дата звернення: 11.04.2023)
24. Посилання на офіційне джерело W3C. URL: <https://www.w3.org/2002/01/spellchecker> (дата звернення: 14.02.2023)
25. Середня заробітна плата розробника. URL: <https://www.work.ua/salary/?search=front-end> (дата звернення: 23.05.2023)
26. Методичні рекомендації до виконання кваліфікаційних робіт здобувачів першого рівня вищої освіти спеціальності 121 Інженерія програмного забезпечення / В.В. Спирінцев, І.М. Удовик, О.С. Шевцова; Д : НТУ «Дніпровська політехніка», 2022. – 60 с.

ЛІСТИНГ ПРОГРАМИ

Index.php

```
<?php get_header(); ?>
```

```
<?php get_footer(); ?>
```

Header.php

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <meta charset="utf-8">
    <meta name="theme-color" content="#3D4356">
    <link rel="icon" type="image/png" href="<?php the_field( 'favicon', 'option' ); ?>">
    <link href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@400;600;700&display=swap"
rel="stylesheet">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/Swiper/5.4.5/css/swiper.min.css">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/gh/fancyapps/fancybox@3.5.7/dist/jquery.fancybox.min.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.2.0/css/font-awesome.min.css">
    <link href="<?php echo get_template_directory_uri(); ?>/assets/css/app.css?v=1675668052785" rel="stylesheet">
    <!-- Google tag (gtag.js) -->
    <script async src="https://www.googletagmanager.com/gtag/js?id=G-7R2YRSXPZ7"></script>
    <script>
      window.dataLayer = window.dataLayer || [];
      function gtag(){dataLayer.push(arguments);}
      gtag('js', new Date());
      gtag('config', 'G-7R2YRSXPZ7');
    </script>
    <?php wp_head(); ?>
  </head>
  <body>
    <div class="preloader">
      <div class="preloader-logo"></div>
      <div class="preloader-line">
        <div class="preloader-loading"></div>
      </div>
    </div>
    <script>
      let preloader = document.querySelector('.preloader');

      if (!!(document.referrer.indexOf('fearless') == -1) || !(location.href.indexOf('localhost') == -1)) {
        preloader.remove();
      }
    </script>
  </body>
</html>
```

```

}

setTimeout(function () {
    if(preloader) {
        setTimeout(function () {
            preloader.classList.add('preloader--hide');
        }, 2000);

        setTimeout(function () {
            preloader.style.display = 'none';
        }, 3500);
    }
}, 500);
</script>
<header class="header" aria-label="Main">
<div class="center-wrap">
<div class="header__wrap">
<a class="main-content" href="#main">Skip to main content</a>
<?php if (is_front_page()) {?>
<a class="header_logo single-anchors img-wrap" href="#home">
<?php } else {?>
<a class="header__logo img-wrap" href="<?php echo home_url(); ?>">
<?php }?>
<div>
Home"></div></a>
<nav class="header__nav mobile--hide">

<?php if ( have_rows( 'header_menu', 'option' ) ) : ?>
<ul class="header__nav-list" role="menu">
<?php while ( have_rows( 'header_menu', 'option' ) ) : the_row(); ?>
<?php $item = get_sub_field( 'item' ); ?>
<?php if ( $item ) : ?>
<?php if ( have_rows( 'submenu' ) ) : ?>
<li class="header__nav-item header__nav-item--drop" role="menuitem" aria-haspopup="true" aria-expanded="false"
tabindex="0"><span class="header__nav-link"><?php echo esc_html( $item['title'] ); ?></span>
<ul class="header__drop header__drop--width header__drop--js">
<?php while ( have_rows( 'submenu' ) ) : the_row(); ?>
<?php $subitem = get_sub_field( 'subitem' ); ?>
<?php if ( $subitem ) : ?>
<li class="header__drop-item"><a class="header__nav-link" href="<?php echo esc_url( $subitem['url'] ); ?>"
target="<?php echo esc_attr( $subitem['target'] ); ?>"><?php echo esc_html( $subitem['title'] ); ?></a></li>
<?php endif; ?>
<?php endwhile; ?>
</ul>
</li>
<?php else : ?>
<li class="header__nav-item" role="menuitem"><a class="header__nav-link" href="<?php echo esc_url( $item['url'] );
?>" target="<?php echo esc_attr( $item['target'] ); ?>"><?php echo esc_html( $item['title'] ); ?></a></li>
<?php endif; ?>
<?php endif; ?>
<?php endwhile; ?>
</ul>

```

```

<?php endif; ?>
    </nav>
    <div class="header__search">
    </div>
    <?php $header_button = get_field( 'header_button', 'option' ); ?>
<?php if( $header_button ) : ?>
    <a class="header__btn btn mobile--hide" href="<?php echo esc_url( $header_button[url] ); ?>" target="<?php echo
esc_attr( $header_button[target] ); ?>"><?php echo esc_html( $header_button[title] ); ?></a>
<?php endif; ?>
    <button class="header__burger desktop--hide" aria-label="Main Menu" aria-expanded="false">
    <div></div>
    <div></div>
    </button>
</div>
</div>
</header>
<div id="scrollbar">
    <main id="main">

```

Footer.php

```

</main>
<footer class="footer">
<div class="footer_top">
<div class="footer_bg"></div>
<div class="center-wrap">
<div class="footer_top-row">
    <nav class="footer__nav" aria-label="Footer">
<?php if( have_rows( 'footer_menu', 'option' ) ) : ?>
    <ul class="footer__nav-list" role="list">
    <?php while ( have_rows( 'footer_menu', 'option' ) ) : the_row(); ?>
    <?php $item = get_sub_field( 'item' ); ?>
    <?php if( $item ) : ?>
        <li class="footer__nav-item" role="listitem"> <a class="footer__nav-link" href="<?php echo esc_url( $item[url] );
?>" target="<?php echo esc_attr( $item[target] ); ?>"><?php echo esc_html( $item[title] ); ?></a></li>
    <?php endif; ?>
    <?php endwhile; ?>
    </ul>
<?php endif; ?>
</nav>
    <div class="footer_top-box"><a class="footer_top-email" href="mailto:<?php the_field( 'footer_email', 'option' ); ?>"
target="_blank"><?php the_field( 'footer_email', 'option' ); ?></a>
    <div class="footer__socials">
    <?php if( have_rows( 'footer_social', 'option' ) ) : ?>
    <ul class="socials" role="list">
    <?php while ( have_rows( 'footer_social', 'option' ) ) : the_row(); ?>
    <?php if( get_sub_field( 'type' ) === 'instagram' ) { ?>
    <li class="socials__item" role="listitem"><a class="socials__link" href="<?php the_sub_field( 'link' ); ?>"
target="_blank">
        <svg viewBox="0 0 24 24" fill="none" xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Instagram">
        <path fill-rule="evenodd" clip-rule="evenodd" d="M4.59961 9.99998C4.59961 7.75014 4.59961 6.62522
5.17256 5.83662C5.3576 5.58194 5.58157 5.35796 5.83625 5.17292C6.62485 4.59998 7.74977 4.59998 9.99961
4.59998H13.9996C16.2494 4.59998 17.3744 4.59998 18.163 5.17292C18.4176 5.35796 18.6416 5.58194 18.8267
5.83662C19.3996 6.62522 19.3996 7.75014 19.3996 9.99997V14C19.3996 16.2498 19.3996 17.3747 18.8267

```

```
18.1633C18.6416 18.418 18.4176 18.642 18.163 18.827C17.3744 19.4 16.2494 19.4 13.9996 19.4H9.99961C7.74977
19.4 6.62485 19.4 5.83625 18.827C5.58157 18.642 5.3576 18.418 5.17256 18.1633C4.59961 17.3747 4.59961 16.2498
4.59961 14V9.99998ZM11.9995 8.08233C9.83601 8.08233 8.08188 9.83646 8.08188 12C8.08188 14.1635 9.83601
15.9176 11.9995 15.9176C14.163 15.9176 15.9172 14.1635 15.9172 12C15.9172 9.83646 14.163 8.08233 11.9995
8.08233ZM11.9995 14.4485C10.6499 14.4485 9.551 13.3496 9.551 12C9.551 10.6494 10.6499 9.55145 11.9995
9.55145C13.3492 9.55145 14.4481 10.6494 14.4481 12C14.4481 13.3496 13.3492 14.4485 11.9995
14.4485ZM16.7331 7.78851C16.7331 8.07682 16.4994 8.31054 16.2111 8.31054C15.9228 8.31054 15.6891 8.07682
15.6891 7.78851C15.6891 7.5002 15.9228 7.26648 16.2111 7.26648C16.4994 7.26648 16.7331 7.5002 16.7331
7.78851Z"></path>
</svg><span class="socials__item-text">Opening in a new window</span></a></li>
<?php } elseif(get_sub_field( 'type' ) === 'facebook') {?>
<li class="socials__item" role="listitem"><a class="socials__link" href="<?php the_sub_field( 'link' ); ?>"
target="_blank">
<svg viewBox="0 0 24 24" fill="none" xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Facebook">
<path d="M12.9819 20.3999C13.2028 20.3999 13.3819 20.2208 13.3819 19.9999V13.1366C13.3819 12.9157
13.561 12.7366 13.7819 12.7366H15.5651C15.7665 12.7366 15.9365 12.5868 15.9619 12.387L16.2398
10.2005C16.2702 9.96152 16.0839 9.75009 15.843 9.75009H13.7819C13.561 9.75009 13.3819 9.57101 13.3819
9.35009V7.8433C13.3819 6.97865 13.6186 6.38935 14.8409 6.38935L16 6.38881C16.2208 6.3887 16.3998 6.20965
16.3998 5.98881V4.06735C16.3998 3.86703 16.2518 3.6987 16.0523 3.68118C15.6242 3.64359 14.9168 3.59998
14.1282 3.59998C11.8807 3.59998 10.3419 4.9917 10.3419 7.54767V9.35017C10.3419 9.57109 10.1628 9.75018
9.94191 9.75018H8.1998C7.97889 9.75018 7.7998 9.92926 7.7998 10.1502V12.3367C7.7998 12.5576 7.97889
12.7367 8.1998 12.7367H9.94183C10.1627 12.7367 10.3418 12.9158 10.3418 13.1367V20C10.3418 20.2209 10.5209
20.4 10.7418 20.4L12.9819 20.3999Z"></path>
</svg><span class="socials__item-text">Opening in a new window</span></a></li>
<?php } elseif(get_sub_field( 'type' ) === 'twitter') {?>
<li class="socials__item" role="listitem"><a class="socials__link" href="<?php the_sub_field( 'link' ); ?>"
target="_blank">
<svg viewBox="0 0 24 24" fill="none" xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Twitter">
<path d="M22 5.89789C21.2645 6.21861 20.4744 6.43531 19.6438 6.53326C20.5008 6.0286 21.142 5.23432
21.4477 4.29863C20.6425 4.7693 19.7612 5.1006 18.8422 5.27812C18.2242 4.62877 17.4057 4.19836 16.5136
4.05374C15.6216 3.90911 14.706 4.05834 13.909 4.47828C13.1119 4.89821 12.4781 5.56535 12.1058
6.37611C11.7336 7.18688 11.6437 8.0959 11.8503 8.96206C10.2187 8.88144 8.62259 8.46412 7.16553
7.73718C5.70847 7.01023 4.42301 5.98991 3.39258 4.74244C3.04025 5.34053 2.83766 6.03398 2.83766
6.7725C2.83727 7.43734 3.00364 8.09199 3.32201 8.67838C3.64038 9.26477 4.10091 9.76476 4.66273
10.134C4.01117 10.1136 3.37397 9.94032 2.80419 9.62863V9.68064C2.80413 10.6131 3.13189 11.5169 3.73186
12.2386C4.33184 12.9603 5.16707 13.4555 6.09583 13.6402C5.4914 13.8012 4.85769 13.8249 4.24258
13.7096C4.50462 14.5119 5.01506 15.2135 5.70243 15.7162C6.3898 16.2188 7.21969 16.4974 8.07593
16.5128C6.62242 17.6357 4.82735 18.2448 2.97948 18.2421C2.65215 18.2422 2.32509 18.2234 2 18.1858C3.87569
19.3726 6.05914 20.0024 8.28909 20C15.8378 20 19.9644 13.8474 19.9644 8.51132C19.9644 8.33796 19.96 8.16286
19.9521 7.9895C20.7548 7.41825 21.4476 6.71087 21.9982 5.90049L22 5.89789Z"></path>
</svg><span class="socials__item-text">Opening in a new window</span></a></li>
<?php } elseif(get_sub_field( 'type' ) === 'youtube') {?>
<li class="socials__item" role="listitem"><a class="socials__link" href="<?php the_sub_field( 'link' ); ?>"
target="_blank">
<svg viewBox="0 0 24 24" fill="none" xmlns="http://www.w3.org/2000/svg" role="img" aria-label="YouTube">
<path d="M21.5781 7.18571C21.4643 6.76289 21.2414 6.37737 20.9318 6.06774C20.6222 5.75811 20.2367
5.53524 19.8139 5.42143C18.2568 5 12 5 5 5 5 5.74319 5 4.18613 5.42143C3.76333 5.53524 3.37783 5.75811
3.06822 6.06774C2.75861 6.37737 2.53575 6.76289 2.42194 7.18571C2.13121 8.77376 1.99012 10.3856 2.00054
12C1.99012 13.6144 2.13121 15.2262 2.42194 16.8143C2.53575 17.2371 2.75861 17.6226 3.06822 17.9323C3.37783
18.2419 3.76333 18.4648 4.18613 18.5786C5.74319 19 12 19 19 19 19 18.2568 19 19.8139 18.5786C20.2367
18.4648 20.6222 18.2419 20.9318 17.9323C21.2414 17.6226 21.4643 17.2371 21.5781 16.8143C21.8688 15.2262
22.0099 13.6144 21.9995 12C22.0099 10.3856 21.8688 8.77376 21.5781 7.18571ZM10.0001 15V9L15.1927
12L10.0001 15Z"></path>
```

```

</svg><span class="socials__item-text">Opening in a new window</span></a></li>
<?php } elseif(get_sub_field('type') === 'linkedin') {?>
<li class="socials__item" role="listitem"><a class="socials__link" href="<?php the_sub_field('link'); ?>"
target="_blank">
<svg viewBox="0 0 24 24" fill="none" xmlns="http://www.w3.org/2000/svg" role="img" aria-label="LinkedIn">
<path d="M18.3878 18.3367C17.5776 18.3175 16.7867 18.3175 15.9766 18.3367C15.803 18.3367 15.7644
18.2982 15.7644 18.1246C15.7644 16.6971 15.7644 15.2504 15.7644 13.823C15.7644 13.4951 15.7451 13.1672
15.6487 12.8585C15.3593 11.8555 14.202 11.489 13.3725 12.1641C12.9288 12.5113 12.7552 12.9935 12.7552
13.5722C12.7552 14.9225 12.7552 16.2728 12.7552 17.623C12.7552 17.7966 12.7359 17.9702 12.7552
18.1631C12.7728 18.3034 12.7106 18.3481 12.5977 18.3405C12.5741 18.3389 12.5505 18.3367 12.5268
18.3367C11.7097 18.3367 10.9109 18.3367 10.0933 18.3367C9.93897 18.3367 9.90039 18.2982 9.90039
18.1438C9.91968 16.9286 9.91968 15.7134 9.91968 14.4788C9.91968 12.9743 9.91968 11.4697 9.90039
9.98439C9.90039 9.81078 9.93897 9.7722 10.0933 9.7722C10.9227 9.7722 11.7329 9.7722 12.5623 9.7722C12.7167
9.7722 12.7552 9.81078 12.7552 9.9651C12.7552 10.2362 12.7552 10.5072 12.7552 10.8175C12.7552 10.866 12.8153
10.8888 12.8495 10.8546V10.8546C12.8519 10.8522 12.8527 10.851 12.8546 10.8482C13.6067 9.75212 14.6665
9.44453 15.9187 9.63717C17.3654 9.86865 18.2913 10.8717 18.5228 12.4149C18.5807 12.7814 18.6 13.1479 18.6
13.5144C18.6 15.0575 18.6 16.5814 18.6 18.1246C18.6 18.2789 18.5614 18.3367 18.3878 18.3367Z"></path>
<path d="M8.35704 14.0544C8.35704 15.4047 8.35704 16.755 8.35704 18.1052C8.35704 18.2788 8.31846
18.3367 8.14486 18.3367C7.3347 18.3174 6.52454 18.3367 5.71438 18.3367C5.56006 18.3367 5.52148 18.2981
5.52148 18.1438C5.52148 15.424 5.52148 12.6849 5.52148 9.96505C5.52148 9.83002 5.56006 9.77216 5.71438
9.77216C6.54383 9.77216 7.37328 9.77216 8.20273 9.77216C8.37633 9.77216 8.39562 9.83002 8.39562
9.98434C8.35704 11.3346 8.35704 12.6849 8.35704 14.0544Z"></path>
<path d="M8.49227 7.53457C8.29938 8.30615 7.50851 8.74981 6.6019 8.59549C5.63742 8.44117 5.09732
7.49599 5.44453 6.57009C5.676 5.99141 6.2354 5.64419 6.94911 5.66348C8.02932 5.64419 8.74304 6.51222 8.49227
7.53457Z"></path>
</svg><span class="socials__item-text">Opening in a new window</span></a></li>
<?php }?>
<?php endwhile; ?>
</ul>
<?php endif; ?>
</div>
</div>
<div class="footer__comp img-wrap desktop--hide">
<div>
<?php $certificate_image = get_field('certificate_image', 'option'); ?>
<?php if( $certificate_image ): ?>
<picture>"></picture>
<?php endif; ?>
</div>
</div>
</div>
<div class="footer__top-row">
<div class="footer__top-left">
<?php if(is_front_page()) {?>
<a class="header__logo single-anchors img-wrap" href="#home">
<?php } else {?>
<a class="footer__logo img-wrap" href="<?php echo home_url(); ?>">
<?php }?>
<div>
Home"></div></a>
<p><?php the_field('site_title', 'option'); ?></p>
<div class="footer__top-info"><?php the_field('footer_address', 'option'); ?></div>

```



```

</div>
<div class="footer__sing">
  <div class="footer__sing-title"><?php the_field( 'newsletter_title', 'option' ); ?></div>
  <div class="footer__sing-info"><?php the_field( 'newsletter_text', 'option' ); ?></div>
</div>
<?php $item = get_field( 'email', 'option' ); ?>
<?php if ( $item ) : ?>
  <a class="footer__btn btn btn--empty" href="<?php echo esc_url( $item[url] ); ?>" target="<?php echo esc_attr(
$item[target] ); ?>"><?php echo esc_html( $item[title] ); ?></a>
<?php endif; ?>
<div class="footer__comp img-wrap mobile--hide">
  <div>
    <?php $certificate_image = get_field( 'certificate_image', 'option' ); ?>
<?php if ( $certificate_image ) : ?>
  <picture>"></picture>
  <?php endif; ?>
</div>
</div>
</div>
</div>
</div>
<div class="footer__bottom mobile--hide">
<div class="center-wrap">
<p class="footer__bottom-copy"><?php the_field( 'copyright', 'option' ); ?></p>
</div>
</div>
</footer>
</div>
<div class="mobile-menu">
<div class="mobile-menu__wrap">
<button class="mobile-menu__act mobile-menu__act--close" aria-label="Close menu">
<svg viewBox="0 0 24 24" fill="none" xmlns="http://www.w3.org/2000/svg">
  <path d="M18 7.05L16.95 6L12 10.95L7.05 6L6 7.05L10.95 12L6 16.95L7.05 18L12 13.05L16.95 18L18 16.95L13.05
12L18 7.05Z"></path>
</svg>
</button>
<div class="mobile-menu__body">
<nav class="mobile-menu__nav">
  <?php if ( have_rows( 'header_menu', 'option' ) ) : ?>
  <ul class="mobile-menu__nav-list" role="list">
    <?php while ( have_rows( 'header_menu', 'option' ) ) : the_row(); ?>
    <?php $item = get_sub_field( 'item' ); ?>
    <?php if ( $item ) : ?>
      <?php if ( have_rows( 'submenu' ) ) : ?>
        <li class="mobile-menu__nav-item mobile-menu__nav-item--drop" role="listitem" aria-haspopup="true" aria-
expanded="false" tabindex="0">
          <p><?php echo esc_html( $item[title] ); ?></p>
          <ul class="mobile-menu__drop mobile-menu__drop--js" role="list">
            <?php while ( have_rows( 'submenu' ) ) : the_row(); ?>
            <?php $subitem = get_sub_field( 'subitem' ); ?>
            <?php if ( $subitem ) : ?>

```

```

        <li class="mobile-menu__drop-item" role="listitem"><a class="mobile-menu__drop-link" href="<?php echo
esc_url( $subitem['url'] ); ?>" target="<?php echo esc_attr( $subitem['target'] ); ?>" role="button"><?php echo esc_html(
$subitem['title'] ); ?></a></li>
        <?php endif; ?>
    <?php endwhile; ?>
    </ul>
</li>
<?php else : ?>
    <li class="mobile-menu__nav-item" role="listitem"><a class="mobile-menu__nav-link" href="<?php echo
esc_url( $item['url'] ); ?>" target="<?php echo esc_attr( $item['target'] ); ?>" role="button"><?php echo esc_html(
$item['title'] ); ?></a></li>
    <?php endif; ?>
<?php endif; ?>
<?php endwhile; ?>
</ul>
<?php endif; ?>

</nav>
<?php $header_button = get_field( 'header_button', 'option' ); ?>
<?php if ( $header_button ) : ?>
    <a class="btn mobile-menu__btn" href="<?php echo esc_url( $header_button['url'] ); ?>" target="<?php echo esc_attr(
$header_button['target'] ); ?>"><?php echo esc_html( $header_button['title'] ); ?></a>
<?php endif; ?>
</div>
</div>
</div>
<div class="popup-wrap" id="thank-you">
<div class="popup">
<div class="popup__close" onclick="close_popup('thank-you')">
<div></div>
<div></div>
</div>
<div class="popup__inner">
<div class="popup__simple-title simple-title"><?php the_field( 'thaks_message_title', 'option' ); ?></div>
<div class="popup__content content">
    <p><?php the_field( 'thaks_message_text', 'option' ); ?></p>
</div>
</div>
</div>
</div>
<?php wp_footer(); ?>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script src="https://cdn.jsdelivr.net/gh/fancyapps/fancybox@3.5.7/dist/jquery.fancybox.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/Swiper/5.4.5/js/swiper.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.5.1/gsap.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.5.1/ScrollTrigger.min.js"></script>
<script src="<?php echo get_template_directory_uri(); ?>/js/send.js?v=1675668051324"></script>
<script src="<?php echo get_template_directory_uri(); ?>/js/app.js?v=1675668051324"></script>
<script>
$(window).bind('load', function() {
    if(window.location.hash) {
        let hash = window.location.hash.substring(1);
        if($('.training__top-item.' + hash).length > 0) {

```

```

        document.querySelector('.training__top-item.' + hash).click();
        setTimeout(function () {document.querySelector('.training__top-item.' + hash).click()}, 500);
    }
}
});
</script>
<style>
.footer__comp {
    max-width: 14rem;
}
</style>
</body>
</html>

```

Function.php

```

<?php
require get_template_directory() . '/inc/add_ajax.php';
require get_template_directory() . '/inc/ajax_function.php';
require get_template_directory() . '/inc/options_page.php'; //добавляем страницу настроек темы для ACF
require get_template_directory() . '/inc/remove_unnecessary.php'; //удаляем все ненужные теги скрипты и стили,
которые вордпресс сам выводит
require get_template_directory() . '/inc/hide_the_face_in_the_menb.php'; //удаляем все ненужные пункты из админки
require get_template_directory() . '/inc/hide_editor.php'; //удаляем редактор блоков
function superbay_setup() {
    add_theme_support('html5', array('comment-list', 'comment-form', 'search-form', 'gallery', 'caption', 'style', 'script'));
    add_theme_support('title-tag');
}
add_action('after_setup_theme', 'superbay_setup');

function register_styles_scripts() {
    wp_enqueue_style('superbay-style', get_stylesheet_uri());
}
add_action('wp_enqueue_scripts', 'register_styles_scripts');
add_action('admin_head', 'Custom_styles_for_admin_panel');
function Custom_styles_for_admin_panel() {
    print '<style>
.update-nag.notice,
#wp-admin-bar-updates,
.plugin-update-tr,
span.update-plugins,
.subsubsub li.upgrade,
#menu-dashboard > ul.wp-submenu > li:last-child {
    display: none !important;
}
</style>';
}

```

Частини теми, з якого складається сайт

accessibility.php

```

<?php if (get_field( 'accessibility_text' ) !== '' ) {?>
<section class="accessibility">

```

```

<div class="accessibility__decor img-wrap">
<div></div>
</div>
<div class="center-wrap">
<div class="accessibility__wrap content">
  <?php the_field( 'accessibility_text' ); ?>
</div>
</div>
</section>
<?php } ?>

```

advantage.php

```

<div class="advantage at-block">
  <div class="advantage__bg"></div>
  <div class="advantage__bg advantage__bg--bottom"></div>
<div class="center-wrap">
<div class="advantage__wrap">
<div class="advantage__left img-wrap at-element" data-at-name="atCustom">
  <div>
    <?php if( get_field( 'advantage_image' ) ): ?>
    
    <?php endif?>
  </div>
</div>
<div class="advantage__right">
  <div class="advantage__content content at-element" data-at-name="atText">
    <?php the_field( 'advantage_text' ); ?>
  </div>
  <?php $advantage_button = get_field( 'advantage_button' ); ?>
  <?php if( $advantage_button ): ?>
    <a class="advantage__btn btn at-element" data-at-name="atCustom" href="<?php echo esc_url(
$advantage_button['url'] ); ?>" target="<?php echo esc_attr( $advantage_button['target'] ); ?>"><?php echo esc_html(
$advantage_button['title'] ); ?></a>
  <?php endif; ?>
</div>
</div>
</div>
</div>

```

Advisory.php

```

<section class="advisory">
  <?php if( have_rows( 'advisory_content' ) ): ?>
<div class="center-wrap">
<div class="advisory__wrap">
<div class="advisory__content content">
  <?php while ( have_rows( 'advisory_content' ) ): the_row(); ?>
    <?php if( get_sub_field( 'type' ) == 'text' ) { ?>
<?php the_sub_field( 'text' ); ?>
    <?php } elseif( get_sub_field( 'type' ) == 'list' ) { ?>
    <?php if( have_rows( 'list' ) ): ?>
    <ul role="list">
    <?php while ( have_rows( 'list' ) ): the_row(); ?>
      <li role="listitem"><?php the_sub_field( 'item' ); ?></li>
    <?php endwhile; ?>

```

```

</ul>
<?php endif; ?>
<?php } ?>
<?php endwhile; ?>
</div>
</div>
</div>
<?php endif; ?>
</section>

```

Audit.php

```

<section class="audit">
<div class="center-wrap">
<div class="audit_wrap">
<h2 class="audit__simple-title simple-title simple-title--left"><?php the_field( 'audit_title' ); ?></h2>
<div class="audit__content content">
<?php the_field( 'audit_text' ); ?>
</div>
<div class="audit__list">
<h2 class="audit__simple-title simple-title"><?php the_field( 'audit_list_title' ); ?></h2>
<?php if( have_rows( 'audit_list' ) ) : ?>
<ul class="audit__list-wrap" role="list">
<?php while ( have_rows( 'audit_list' ) ) : the_row(); ?>
<li class="audit__list-item" role="listitem">
<div class="audit__list-content content">
<p><?php the_sub_field( 'text' ); ?></p>
</div>
</li>
<?php endwhile; ?>
</ul>
<?php endif; ?>
</div>
</div>
</div>
<?php if( have_rows( 'audit_boxes_list' ) ) : ?>
<?php while ( have_rows( 'audit_boxes_list' ) ) : the_row(); ?>
<div class="audit__box"><?php if( (isset(get_sub_field('background')[0])) && (get_sub_field('background')[0] === 'yes')) {
?> audit_box--bg<?php } ?>>
<div class="center-wrap">
<div class="audit__box-wrap">
<?php if( have_rows( 'left_side' ) ) : ?>
<div class="audit__box-left content">
<?php while ( have_rows( 'left_side' ) ) : the_row(); ?>

<?php if( get_sub_field( 'type' ) === 'text' ) { ?>
<?php the_sub_field( 'text' ); ?>
<?php } elseif( get_sub_field( 'type' ) === 'list' ) { ?>
<?php if( have_rows( 'list' ) ) : ?>
<ul role="list">
<?php while ( have_rows( 'list' ) ) : the_row(); ?>
<li role="listitem"><?php the_sub_field( 'item' ); ?></li>
<?php endwhile; ?>
</ul>
<?php endif; ?>

```

```

        <?php } ?>
    <?php endwhile; ?>
</div>
<?php endif; ?>
<div class="audit__box-right img-wrap">
    <div>
        <?php if ( get_sub_field( 'image' ) ) : ?>
            <picture></picture>
        <?php endif ?>
    </div>
</div>
</div>
</div>
</div>
</div>
<?php endwhile; ?>
<?php endif; ?>

<?php if ( have_rows( 'audit_form' ) ) : ?>
    <?php while ( have_rows( 'audit_form' ) ) : the_row(); ?>
<div class="audit_form">
<div class="center-wrap">
<div class="audit__form-wrap">
    <h2 class="audit__form-title simple-title"><?php the_sub_field( 'title' ); ?></h2>
    <div class="audit__form-content content">
        <?php the_sub_field( 'text' ); ?>
    </div>
    <div class="audit__form-row">
        <div class="audit__form-content content">
            <?php the_sub_field( 'left_side' ); ?>
        </div>
        <div class="audit__form-img img-wrap">
            <div>
                <?php if ( get_sub_field( 'image' ) ) : ?>
                    <picture></picture>
                <?php endif ?>
            </div>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
<?php endwhile; ?>
<?php endif; ?>
</section>

```

Blog.hph

```

<div class="blog">
<div class="center-wrap">
<div class="blog__box">
    <?php $posts = get_posts([
        'post_status' => 'publish',
        'numberposts' => -1,
        'fields' => 'ids'
    ]); ?>

```

```

<ul class="blog__box-top" role="list">
  <?php $iterator = 0; ?>
  <?php foreach ($posts as $value) {?>
    <?php $iterator++; ?>
    <?php if ($iterator === 1) {?>
<li class="blog__box-item blog__box-item--big" role="listitem">
  <div class="blog__box-wrap">
    <div class="blog__box-img img-wrap img-parallax">
      <div>
        <?php if (get_field('main_image', $value)) {?>
          <picture></picture>
        <?php } elseif (get_field('header_image', $value)) {?>
          <picture></picture>
        <?php } else {?>
          <picture></picture>
        <?php } ?>
      </div>
    </div>
    <div class="blog__box-info">
      <div class="blog__box-date"><?php echo get_the_date('F j, Y', $value); ?> </div>
      <h2 class="blog__box-title" id="title<?php echo $iterator; ?>"><?php echo get_the_title($value); ?></h2>
      <div class="blog__box-content content">
        <p><?php echo get_field('post_text_title', $value); ?></p>
      </div><a class="blog__box-read" href="<?php echo get_permalink($value); ?>" aria-labelledby="title1<?php echo
$iterator; ?> title<?php echo $iterator; ?>">Read more</a>
    </div>
  </div>
</li>
<?php } else {?>
  <?php break; ?>
<?php } ?>
<?php } ?>
<?php $iterator = 0; ?>
<?php foreach ($posts as $value) {?>
  <?php $iterator++; ?>
  <?php if (($iterator < 4) && ($iterator > 1)) {?>
<li class="blog__box-item" role="listitem">
  <div class="blog__box-wrap blog__box-wrap--small">
    <div class="blog__box-img img-wrap img-parallax">
      <div>
        <?php if (get_field('main_image', $value)) {?>
          <picture></picture>
        <?php } elseif (get_field('header_image', $value)) {?>
          <picture></picture>
        <?php } else {?>
          <picture></picture>
        <?php } ?>
      </div>
    </div>
    <div class="blog__box-info">
      <div class="blog__box-date"><?php echo get_the_date('F j, Y', $value); ?> </div>

```

```

    <h2 class="blog_box-title" id="title"<?php echo $iterator; ?>><?php echo get_the_title($value); ?></h2>
    <div class="blog_box-content content">
        <p><?php echo get_field('post_text_title', $value); ?></p>
    </div><a class="blog_box-read" href="<?php echo get_permalink($value); ?>" aria-labeledby="title 1"<?php echo
$iterator; ?> title"<?php echo $iterator; ?>">Read more</a>
    </div>
</div>
</li>
<?php } elseif ($iterator === 4) { ?>
    <?php break; ?>
<?php } ?>
<?php } ?>
</ul>
<ul class="blog_box-bottom" role="list">
    <?php $iterator = 0; ?>
    <?php foreach ($posts as $value) { ?>
        <?php $iterator++; ?>
        <?php if (($iterator > 3) && ($iterator <= 5)) { ?>
<li class="blog_row" role="listitem">
    <div class="blog_row-date">
        <p><?php echo get_the_date('j', $value); ?></p><span><?php echo get_the_date('F', $value); ?></span>
    </div>
    <div class="blog_row-img img-wrap img-parallax">
        <div>
            <?php if (get_field('main_image', $value)) { ?>
                <picture></picture>
            <?php } elseif (get_field('header_image_mobile', $value)) { ?>
                <picture></picture>
            <?php } else { ?>
                <picture></picture>
            <?php } ?>
        </div>
    </div>
    <div class="blog_row-info">
        <h2 class="blog_row-title" id="title"<?php echo $iterator; ?>><?php echo get_the_title($value); ?></h2>
        <div class="blog_row-content content">
            <p><?php echo get_field('post_text_title', $value); ?></p>
        </div><a class="blog_box-read" href="<?php echo get_permalink($value); ?>" aria-labeledby="title 1"<?php echo
$iterator; ?> title"<?php echo $iterator; ?>">Read more</a>
        </div>
    </li>
<?php } elseif ($iterator > 5) { ?>
<li class="blog_row hide_item_press" role="listitem">
    <div class="blog_row-date">
        <p><?php echo get_the_date('j', $value); ?></p><span><?php echo get_the_date('F', $value); ?></span>
    </div>
    <div class="blog_row-img img-wrap img-parallax">
        <div>
            <?php if (get_field('main_image', $value)) { ?>
                <picture></picture>
            <?php } elseif (get_field('header_image_mobile', $value)) { ?>

```



```

        <picture></picture>
    <?php } else { ?>
        <picture></picture>
    <?php } ?>
</div>
</div>
<div class="blog__row-info">
    <h2 class="blog__row-title" id="title"<?php echo $iterator; ?>"><?php echo get_the_title($value); ?></h2>
    <div class="blog__row-content content">
        <p><?php echo get_field('post_text_title', $value); ?></p>
    </div><a class="blog__box-read" href="<?php echo get_permalink($value); ?>" aria-label="title 1"<?php echo
$iterator; ?> title"><?php echo $iterator; ?>">Read more</a>
</div>
</li>
<?php } ?>
<?php } ?>
<div class="blog__btn"> <a class="btn" href="#" onclick="loadDesktopPosts(event);">Load more</a></div>
</ul>
</div>
</div>
</div>
<script>
function loadDesktopPosts(event) {
    event.preventDefault();
    for (let i = 0; i < 3; i++) {
        if (document.querySelector('.hide_item_press')) {
            document.querySelector('.hide_item_press').classList.remove('hide_item_press');
        } else {
            document.querySelector('div.blog__btn').remove();
        }
    }
}
</script>

```

Clients.php

```

<section class="clients at-block" role="region" aria-role-description="carousel" aria-label="Our Clients">
<div class="center-wrap">
<h2 class="clients__simple-title simple-title at-element" data-at-name="atTitle"><?php the_field('clients_title'); ?></h2>
<?php $total = 0; ?>
<?php if (have_rows('clients_clients')) : ?>
    <?php while (have_rows('clients_clients')) : the_row(); ?>
        <?php $total++; ?>
    <?php endwhile; ?>
<?php endif; ?>
<?php $iterator = 0; ?>
<?php if (have_rows('clients_clients')) : ?>
<div class="clients-slider mobile--hide">
<div class="swiper-container">
    <div class="swiper-wrapper">
        <?php while (have_rows('clients_clients')) : the_row(); ?>
            <?php $iterator++; ?>
            <div class="swiper-slide" aria-hidden="true" role="group" aria-role-description="slide" aria-label="<?php echo $iterator;
?> of <?php echo $total; ?>">

```

```

<div class="clients-slider__img img-wrap">
  <div>
    <picture>
      <?php $logo = get_sub_field( 'logo' ); ?>
      <?php if ( $logo ) : ?>
        ">
        <?php endif; ?>
      </picture>
    </div>
  </div>
</div>
<?php endwhile; ?>
</div>
<div class="clients-slider__contr">
  <div class="swiper-button-prev">
    <svg viewBox="0 0 24 24" fill="none" xmlns="http://www.w3.org/2000/svg">
      <path d="M16 4L8 12L16 20" stroke-width="2" stroke-linecap="round" stroke-linejoin="round"></path>
    </svg>
  </div>
  <div class="swiper-pagination" role="group" aria-label="Choose slides"></div>
  <div class="swiper-button-next">
    <svg viewBox="0 0 24 24" fill="none" xmlns="http://www.w3.org/2000/svg">
      <path d="M8 4L16 12L8 20" stroke-width="2" stroke-linecap="round" stroke-linejoin="round"></path>
    </svg>
  </div>
</div>
</div>
</div>
<?php endif; ?>
<?php if ( have_rows( 'clients_clients' ) ) : ?>
<?php $iterator = 0; ?>
<div class="clients__list desktop--hide">
<?php while ( have_rows( 'clients_clients' ) ) : the_row(); ?>
  <?php $iterator++; ?>
  <div class="clients-slider__img img-wrap">
    <?php $logo = get_sub_field( 'logo' ); ?>
    <div>"></div>
  </div>
<?php endwhile; ?>
</div>
<?php endif; ?>
</div>
</section>

```

Contact.php

```

<section class="contact at-block" id="contact">
  <div class="center-wrap">
    <div class="contact__wrap">
      <div class="contact__left">
        <h2 class="contact__simple-title simple-title simple-title--left at-element" data-at-name="atTitle"><?php the_field(
'contact_title' ); ?></h2>
        <div class="contact__content content at-element" data-at-name="atText">
          <p><?php the_field( 'contact_text' ); ?></p>
        </div>
      </div>
    </div>
  </div>
</section>

```

```

<?php if ( have_rows( 'contact_email' ) ): ?>
<div class="contact_list at-element" data-at-name="atCustom">
  <?php while ( have_rows( 'contact_email' ) ): the_row(); ?>
  <a class="contact_link" href="mailto:<?php the_sub_field( 'email' ); ?>" target="_blank">
  <svg viewBox="0 0 36 36" fill="none" xmlns="http://www.w3.org/2000/svg">
    <path fill-rule="evenodd" clip-rule="evenodd" d="M12.2432 16.5946C10.9787 15.7184 10.2208 15.1914 9.96935
15.014C9.51095 14.7034 9.07822 14.2763 8.6716 13.7328C8.26495 13.1893 8.06152 12.6845 8.06152
12.2186C8.06152 11.6419 8.21495 11.1612 8.52179 10.7767C8.82868 10.3922 9.26684 10.2 9.83621
10.2H26.164C26.6445 10.2 27.0606 10.3737 27.4118 10.7213C27.7631 11.0688 27.9388 11.4866 27.9388
11.9748C27.9388 12.5589 27.7577 13.1172 27.3952 13.6494C27.0329 14.1819 26.5819 14.6368 26.0421
15.0137C23.2618 16.9439 21.5312 18.1456 20.851 18.6187C20.7771 18.6706 20.6198 18.7832 20.3795
18.9571C20.1391 19.131 19.9396 19.2716 19.7806 19.3788C19.6216 19.4859 19.4293 19.606 19.2038
19.7392C18.9784 19.872 18.7656 19.9721 18.566 20.0383C18.3663 20.105 18.1814 20.1381 18.0114
20.1381H18.0002H17.989C17.819 20.1381 17.6341 20.1051 17.4344 20.0383C17.2348 19.9721 17.0221 19.8721
16.7966 19.7392C16.5709 19.6061 16.3787 19.4859 16.2197 19.3788C16.0608 19.2716 15.8611 19.131 15.6207
18.9571C15.3804 18.7834 15.2234 18.6706 15.1494 18.6187C14.4765 18.1456 13.5078 17.4709 12.2432
16.5946ZM21.3057 20.0276C22.5627 19.1179 24.4042 17.8422 26.8295 16.2007C27.2512 15.9124 27.6208 15.5905
27.9389 15.2357V24.0427C27.9389 24.5311 27.7652 24.9486 27.4176 25.2962C27.07 25.6439 26.6521 25.8177
26.1642 25.8177H9.83633C9.34824 25.8177 8.93035 25.6439 8.58283 25.2962C8.23523 24.9487 8.06152 24.5311
8.06152 24.0427V15.2357C8.3869 15.598 8.76034 15.9199 9.18195 16.2007C11.8588 18.0198 13.6964 19.2956
14.6948 20.0276C15.1162 20.338 15.4583 20.5804 15.7207 20.7541C15.9832 20.9279 16.3327 21.1053 16.7689
21.2864C17.2052 21.4678 17.6119 21.5582 17.9891 21.5582H18.0002H18.0115C18.3887 21.5582 18.7953 21.4678
19.2315 21.2864C19.6679 21.1053 20.0171 20.9279 20.2797 20.7541C20.5423 20.5804 20.8842 20.338 21.3057
20.0276Z"></path>
  </svg>
  <p><?php the_sub_field( 'email' ); ?></p></a>
  <?php endwhile; ?>
</div>
<?php endif; ?>
</div>
<div class="contact_right">
  <?php if ( have_rows( 'contact_form' ) ): ?>
  <?php while ( have_rows( 'contact_form' ) ): the_row(); ?>
  <form id="form_contacts" class="inputs" action="">
  <fieldset>
    <legend class="contact_right-title"><?php the_sub_field( 'title' ); ?> <br> <?php the_sub_field( 'title_second' );
?></legend>
    <div class="inputs__field-wrap">
      <input class="inputs__field" type="text" id="text" aria-required="true" name="firstname" aria-describedby="text-error"
autocomplete="name" required="" minlength="2">
      <label class="inputs__content" for="text"><?php the_sub_field( 'name_field' ); ?></label>
      <label class="inputs__field-error" id="text-error" for="">The text field is mandatory.</label>
    </div>
    <div class="inputs__field-wrap">
      <input class="inputs__field" type="text" id="textCompany" name="company" aria-describedby="company-error"
name="text" aria-required="true" autocomplete="organization" required="" minlength="2">
      <label class="inputs__content" for="textCompany"><?php the_sub_field( 'company_field' ); ?></label>
    </div>
    <div class="inputs__field-wrap">
      <input class="inputs__field" type="email" id="email" name="email" aria-describedby="mail-error" aria-
required="true" autocomplete="email" required="">
      <label class="inputs__content" for="email"><?php the_sub_field( 'email_field' ); ?></label>
      <label class="inputs__field-error" id="mail-error">The email field is mandatory.</label>
    </div>
  </fieldset>
  </form>
</div>

```

```

</div>
<div class="inputs__field-wrap inputs__field-wrap--textarea">
  <textarea class="inputs__field" name="message" id="textarea" aria-required="true" required></textarea>
  <label class="inputs__content" for="textarea"><?php the_sub_field( 'message_field' ); ?></label>
</div>
<div class="inputs__checkbox">
  <label class="inputs__checkbox-label">
    <input class="inputs__checkbox-input" type="checkbox" checked="checked" required><span
class="inputs__checkbox-content" role="checkbox" aria-labelledby="worldpeace"></span><span
class="inputs__checkbox-labelledby" id="worldpeace">checkbox</span><span class="inputs__checkbox-text"><?php
the_sub_field( 'terms_label' ); ?> </span>

    <?php $terms_link = get_sub_field( 'terms_link' ); ?>
    <?php if ( $terms_link ) : ?>
      <a class="inputs__checkbox-link" href="<?php echo esc_url( $terms_link['url'] ); ?>" target="<?php echo esc_attr(
$terms_link['target'] ); ?>"><?php echo esc_html( $terms_link['title'] ); ?></a>
    <?php endif; ?>
  </label>
</div>
<input type="hidden" name="to" value="<?php the_field( 'forms_admin_email', 'option' ); ?>" />
<div class="inputs__btn">
  <button class="btn" type="submit"><?php the_sub_field( 'button' ); ?></button>
</div>
</fieldset>
</form>
<?php endwhile; ?>
<?php endif; ?>
</div>
</div>
</div>
</section>
<!-- Start of HubSpot Embed Code -->
<script type="text/javascript" id="hs-script-loader" async defer src="//js.hs-scripts.com/2323078.js"></script>
<!-- End of HubSpot Embed Code -->

```

Cooperation.php

```

<section class="cooperation at-block">
  <div class="cooperation__wrap">
    <div class="cooperation__left">
      <h2 class="cooperation__title at-element" data-at-name="atTitle"><?php the_field( 'cooperation_title' ); ?></h2>
      <div class="cooperation__content content at-element" data-at-name="atItemsColumn" data-at-selector=".cooperation__content li, .cooperation__btn" data-at-trigger=".cooperation">
        <?php if ( have_rows( 'cooperation_list' ) ) : ?>
          <ol>
            <?php while ( have_rows( 'cooperation_list' ) ) : the_row(); ?>
              <li><?php the_sub_field( 'item' ); ?></li>
            <?php endwhile; ?>
          </ol>
        <?php endif; ?>
      </div>
      <?php $cooperation_button = get_field( 'cooperation_button' ); ?>
      <?php if ( $cooperation_button ) : ?>
        <a class="cooperation__btn btn" href="<?php echo esc_url( $cooperation_button['url'] ); ?>" target="<?php echo
esc_attr( $cooperation_button['target'] ); ?>"><?php echo esc_html( $cooperation_button['title'] ); ?></a>
      </div>
    </div>
  </div>
</section>

```

```

<?php endif; ?>
</div>
<div class="cooperation__right img-parallax img-wrap">
<div>
    <?php if ( get_field( 'cooperation_image' ) ) : ?>
        <picture></picture>
    <?php endif ?>
</div>
</div>
</div>
</div>
</section>

```

Explainers.php

```

<section class="explainers">
<div class="center-wrap">
<h2 class="explainers__simple-title simple-title"><?php the_field( 'explainers_title' ); ?></h2>
<?php if ( have_rows( 'explainers_list' ) ) : ?>
    <?php $iterator = 0; ?>

<ul class="explainers__list" role="list">
    <?php while ( have_rows( 'explainers_list' ) ) : the_row(); ?>
        <?php $iterator++; ?>
        <li class="explainers__item" role="listitem">
            <div class="explainers__img-wrap img-wrap img-parallax">
                <div>
                    <?php if ( get_sub_field( 'image' ) ) : ?>
                        <picture></picture>
                    <?php endif ?>
                </div>
            </div>
            <div class="explainers__info">
                <h3 class="explainers__info-title" id="linkExplainers<?php echo $iterator; ?>"><?php the_sub_field( 'title' ); ?></h3>
                <?php $button = get_sub_field( 'button' ); ?>
                <?php if ( $button ) : ?>
                    <a class="explainers__info-btn btn btn--empty" href="#<?php echo esc_url( $button['url'] ); ?>" target="#<?php echo esc_attr( $button['target'] ); ?>" aria-label="by="linkExplainers 1<?php echo $iterator; ?> linkExplainers<?php echo $iterator; ?>"><span id="linkExplainers 1<?php echo $iterator; ?>"><?php echo esc_html( $button['title'] ); ?>
                </span></a>
            <?php endif; ?>
        </div>
    </li>
<?php endwhile; ?>

</ul>
<?php endif; ?>

</div>
</section>

```

```

<section class="facilitator-vacancies">
<div class="center-wrap">
<div class="facilitator-vacancies__wrap">
<div class="facilitator-vacancies__content content">

```

```

    <?php the_field( 'facilitator_text' ); ?>
</div>
</div>
</div>
<div class="facilitator-vacancies facilitator-vacancies--bg">
<div class="center-wrap">
<div class="facilitator-vacancies__wrap">
<div class="facilitator-vacancies__content content">
<h2><?php the_field( 'facilitator_list_title' ); ?></h2>
<ul class="facilitator-vacancies__list" role="list">
<?php if ( have_rows( 'facilitator_list' ) ) : ?>
<?php while ( have_rows( 'facilitator_list' ) ) : the_row(); ?>
<li class="facilitator-vacancies__item" role="listitem">
<h3><?php the_sub_field( 'title' ); ?></h3>
<?php the_sub_field( 'text' ); ?>
</li>
<?php endwhile; ?>
<?php endif; ?>
<li class="facilitator-vacancies__item facilitator-vacancies__item--link" role="listitem">
<div class="facilitator-vacancies__item-info">
<p><?php the_field( 'facilitator_list_last_item' ); ?> </p>
<?php $facilitator_list_last_item_link = get_field( 'facilitator_list_last_item_link' ); ?>
<?php if ( $facilitator_list_last_item_link ) : ?>
<a href="<?php echo esc_url( $facilitator_list_last_item_link['url'] ); ?>" target="<?php echo esc_attr(
$facilitator_list_last_item_link['target'] ); ?>"><?php echo esc_html( $facilitator_list_last_item_link['title'] ); ?></a>
<?php endif; ?>
</div>
</li>
</ul>
</div>
</div>
</div>
</div>
<?php if ( have_rows( 'facilitator_process_content' ) ) : ?>
<div class="facilitator-vacancies">
<div class="center-wrap">
<div class="facilitator-vacancies__wrap">
<div class="facilitator-vacancies__content content">
<?php while ( have_rows( 'facilitator_process_content' ) ) : the_row(); ?>
<?php if ( get_sub_field( 'type' ) === 'text' ) { ?>
<?php the_sub_field( 'text' ); ?>
<?php } elseif ( get_sub_field( 'type' ) === 'list' ) { ?>
<?php if ( have_rows( 'list' ) ) : ?>
<ul role="list">
<?php while ( have_rows( 'list' ) ) : the_row(); ?>

```

facilitator-vacancies.php

```
<li role="listitem"><?php the_sub_field( 'item' ); ?>
    <?php if ( have_rows( 'sublist' ) ) : ?>
        <ul role="list">
            <?php while ( have_rows( 'sublist' ) ) : the_row(); ?>
                <li role="listitem"><?php the_sub_field( 'subitem' ); ?></li>
            <?php endwhile; ?>
        </ul>
    <?php endif; ?>
    <?php the_sub_field( 'text_after_sublist' ); ?>
</li>
<?php endwhile; ?>
</ul>
<?php endif; ?>
<?php } ?>
<?php endwhile; ?>
</div>
</div>
</div>
</div>
<?php endif; ?>

<?php if ( have_rows( 'facilitator_apply_content' ) ) : ?>
<div class="facilitator-vacancies facilitator-vacancies--bg">
<div class="center-wrap">
<div class="facilitator-vacancies__wrap">
    <div class="facilitator-vacancies__content content">
        <?php while ( have_rows( 'facilitator_apply_content' ) ) : the_row(); ?>
            <?php if ( get_sub_field( 'type' ) === 'text' ) { ?>
                <?php the_sub_field( 'text' ); ?>
            <?php } elseif ( get_sub_field( 'type' ) === 'list' ) { ?>
                <?php if ( have_rows( 'list' ) ) : ?>
                    <ul class="facilitator-vacancies__list facilitator-vacancies__list--column" role="list">
                        <?php while ( have_rows( 'list' ) ) : the_row(); ?>
                            <li role="listitem"><?php the_sub_field( 'item' ); ?></li>
                        <?php endwhile; ?>
                    </ul>
                <?php endif; ?>
            <?php } ?>
        <?php endwhile; ?>
    </div>
</div>
</div>
</div>
<?php endif; ?>
</section>
```

inclusion-resources.php

```
<section class="inclusion-resources at-block">
<div class="center-wrap">
<h2 class="inclusion-resources__simple-title simple-title at-element" data-at-name="atTitle"><?php the_field(
'inclusion_title' ); ?></h2>
<div class="inclusion-resources__content content at-element" data-at-name="atText">
```

```

<p><?php the_field( 'text' ); ?></p>
</div>
</div>
<?php if ( have_rows( 'list' ) ) : ?>
<div class="inclusion-resources__list">
<?php while ( have_rows( 'list' ) ) : the_row(); ?>
<button class="inclusion-resources__item" tabindex="0">
<div class="inclusion-resources__item-bg"></div>
<div class="inclusion-resources__item-bg inclusion-resources__item-bg--hide"></div>
<div class="inclusion-resources__img inclusion-resources__img--visible img-wrap img-parallax">
<div>
<?php if ( get_sub_field( 'image' ) ) : ?>
<picture></picture>
<?php endif ?>
</div>
</div>
<div class="inclusion-resources__img inclusion-resources__img--hide img-wrap img-parallax">
<div>
<?php if ( get_sub_field( 'hover_image' ) ) : ?>
<picture></picture>
<?php endif ?>
</div>
</div>
<div class="inclusion-resources__info">
<div class="inclusion-resources__title"><?php the_sub_field( 'title' ); ?> <br> <?php the_sub_field( 'title_line' ); ?></div>
<?php $link = get_sub_field( 'link' ); ?>
<?php if ( $link ) : ?>

<a class="inclusion-resources__link" href="<?php echo esc_url( $link['url'] ); ?>" target="<?php echo esc_attr(
$link['target'] ); ?>">
<?php echo esc_html( $link['title'] ); ?>
<svg viewbox="0 0 17 17" fill="none" xmlns="http://www.w3.org/2000/svg">
<path fill-rule="evenodd" clip-rule="evenodd" d="M12.3216 7.26429C12.5316 7.47428 12.3829 7.83333 12.0859
7.83333L2.50016 7.83333C2.13197 7.83333 1.8335 8.1318 1.8335 8.49999C1.8335 8.86818 2.13197 9.16666 2.50016
9.16666L12.086 9.16666C12.3829 9.16666 12.5316 9.52571 12.3217 9.7357L8.69542 13.3619C8.43507 13.6223
8.43507 14.0444 8.69542 14.3047C8.95577 14.5651 9.37788 14.5651 9.63823 14.3047L14.9716 8.9714C15.2319
8.71105 15.2319 8.28894 14.9716 8.0286L9.63823 2.69526C9.37788 2.43491 8.95577 2.43491 8.69542
2.69526C8.43507 2.95561 8.43507 3.37772 8.69542 3.63807L12.3216 7.26429Z" fill="#781E41"></path>
</svg></a>
<?php endif; ?>
</div>
</button>
<?php endwhile; ?>
</div>
<?php endif; ?>
</section>

```

Services.php

```

<section class="services at-block">
<div class="center-wrap">
<h2 class="services__simple-title simple-title at-element" data-at-name="atTitle"><?php the_field( 'services_title' );
?></h2>
<div class="services__wrap">
<?php if ( have_rows( 'services_list' ) ) : ?>

```



```

<?php while ( have_rows( 'services_list' ) ) : the_row(); ?>
  <div class="services__row">
    <h3 class="services__subtitle"><?php the_sub_field( 'title' ); ?></h3>
    <?php if ( have_rows( 'list' ) ) : ?>
      <?php $total = 0; ?>
      <?php while ( have_rows( 'list' ) ) : the_row(); ?>
        <?php $total++; ?>
      <?php endwhile; ?>
      <ul class="services__list"><?php if ($total == 2) { ?> services__list--2<?php } ?>" role="list">
        <?php while ( have_rows( 'list' ) ) : the_row(); ?>
          <li class="services__item" role="listitem">
            <p class="services__item-title"><?php the_sub_field( 'service' ); ?></p>
          </li>
        <?php endwhile; ?>
      </ul>
    <?php endif; ?>
  </div>
<?php endwhile; ?>
<?php endif; ?>
</div>
<?php $button = get_field( 'button' ); ?>
<?php if ( $button ) : ?>
  <div class="services__btn"> <a class="btn" href="<?php echo esc_url( $button['url'] ); ?>" target="<?php echo esc_attr(
  $button['target'] ); ?>"><?php echo esc_html( $button['title'] ); ?></a></div>
<?php endif; ?>
</div>
</section>

```

Всі вихідні файли до проекту, включно з налаштуваннями HubSpot доступно за посиланням [Diploma Zhovnir](#) для співробітників «Dnipro University of Technology»

ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ

ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім'я файлу	Опис
Пояснювальні документи	
Диплом_Жовнір.doc	Пояснювальна записка до кваліфікаційного проекту. Документ Word.
Диплом_Жовнір.pdf	Пояснювальна записка до кваліфікаційного проекту в форматі PDF
Програма	
Diploma_Zhovnir.zip	Архів. Містить код, зображення та стилі програми.
Презентація	
Презентація_диплом Жовнір .ppt	Презентація кваліфікаційної роботи