

ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та
інформаційних технологій
Кафедра інформаційних систем та технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти бакалавр

на тему: **«Проектування вебсайту на екологічну тематику у формі лендінгу
на мові JavaScript»**

Виконав: здобувач вищої освіти
за освітньою програмою
Інформаційні управляючі системи
спеціальності 126 Інформаційні
системи та технології
ступеня вищої освіти бакалавр
групи 126ІСТ_бд_2021
Радченко Владислав Вадимович
Керівник: Одарущенко Олег
Миколайович
Рецензент: Муравльов Володимир
В'ячеславович

Полтава – 2025 року

ВСТУП

Актуальність теми кваліфікаційної роботи розглядається в контексті інформаційного суспільства, коли процеси вирішення різноманітних задач, збору, обробки та передачі великої кількості інформації головним чином здійснюються на основі потужних технологій пошуку даних в мережі інтернет. Технологічні можливості комп'ютерних мереж, особливо інтернет, збільшили швидкість пошуку та обміну інформацією, оскільки тепер легко отримати будь-які дані у відкритому доступі: текстові, візуальні, статистичні, соціальні та безліч інших. Зворотна сторона використання інтернету – необхідність наповнення і представлення якісної актуальної інформації в мережі про найрізноманітніші аспекти людської діяльності на вебплатформах, які побудовані на сучасних вебтехнологіях. Величезна роль відводиться вебсайтам різного типу.

В сучасних умовах загострення екологічних проблем, таких як зміна клімату, забруднення довкілля, знищення біорозмаїття, особливо під час воєнних дій, нераціональне використання природних ресурсів, зростає потреба у підвищенні рівня екологічної свідомості населення. Інформування широкого кола користувачів про сучасні екологічні виклики, можливі шляхи їх розв'язання, а також заохочення до відповідального ставлення до природи є ключовими чинниками сталого розвитку суспільства.

Інформаційний вебсайт екологічної тематики є ефективним інструментом поширення знань, досвіду та практичних порад у сфері охорони довкілля. Такий ресурс може стати корисною платформою як для пересічних громадян, так і для громадських організацій, екологічних ініціатив, освітніх установ та місцевих громад, які прагнуть до екологічних змін.

В умовах цифровізації суспільства доступність якісного та структурованого екологічного контенту в онлайн-середовищі є важливим чинником формування екологічної культури. Саме тому розробка сучасного, інтерактивного та зручного у користуванні вебсайту на екологічну тематику є актуальним і своєчасним завданням.

Метою кваліфікаційної роботи є розроблення інформаційного вебсайту на екологічну тематику із забезпеченням сучасного дизайну на основі широкого спектру вебтехнологій, зокрема мови програмування JavaScript.

Завданнями кваліфікаційної роботи є:

- вивчення інформаційної бази предметної області та користувацьких потреб цільової аудиторії;
- здійснення аналізу технологій розроблення вебсайтів на стороні клієнтів та обґрунтування вибору для подальшої роботи;
- розроблення графічного макету вебсайту і підбір графічного матеріалу з різноманітних джерел;
- програмна реалізація проєкту із забезпеченням інтерактивних та динамічних функцій, у т. ч. за рахунок програмування на JavaScript;
- забезпечення адаптивності сайту для повноцінного перегляду на екранах різного розміру, проведення технічного аудиту і оптимізації вебсайту;
- здійснення економічного аналізу розробки за відомими методами.

Об'єктом дослідження є процес проєктування вебсайту з використанням засобів вебдизайну та вебтехнологій для досягнення заданих параметрів.

Предметом дослідження є комплекс інструментальних засобів і вебтехнологій для дизайну та верстки вебсайтів в середовищах редакторів коду та графічного макетування.

Методами дослідження є: аналітико-синтетичний, емпіричний, графічне проєктування, абстрагування й моделювання, порівняльний, інформаційно-пошуковий, SEO-аналітики вебсайтів, підключення бібліотек JavaScript тощо.

Інформаційна база представлена науковими статтями, популярними публікаціями, матеріалами науково-практичних конференцій всіх рівнів, ресурсами мережі інтернет, узагальненими статистичними даними аналітичних компаній в галузі ІТ, перелік яких подано у списку використаних джерел.

Практична значущість кваліфікаційної роботи полягає в аналізі та систематизації технічних засобів для дизайну і верстки вебсайтів інформаційного призначення, створення оригінального підходу щодо поєднання

відомих технологій у вирішенні конкретного завдання, внесок у методи популяризації екологічної тематики та пріоритетів сталого розвитку для широкого загалу користувачів мережі інтернет, демонстрація успішного теоретичного і прикладного втілення здобутої освіти в області ІТ.

Апробація результатів роботи відбувалася шляхом публікації тез доповіді на науково-практичній конференції за підсумками проходження виробничої практики здобувачів вищої освіти спеціальності 126 Інформаційні системи та технології, яка проходила в ПДАУ 10 жовтня 2024 р.

Структура та обсяг кваліфікаційної роботи. Робота має 56 сторінок, складається зі змісту, вступу, трьох розділів, висновків, списку використаних джерел та додатків. Містить 28 рисунків і 2 таблиці.

РОЗДІЛ 1

РОЗВИТОК ВЕБТЕХНОЛОГІЙ ТА ФУНКЦІЙ ВЕБСАЙТІВ В ІНФОРМАЦІЙНОМУ ЗАБЕЗПЕЧЕННІ ІНТЕРЕСІВ СУСПІЛЬСТВА

1.1 Значення вебсайтів у забезпеченні інформаційних потреб суспільства

Інформаційне суспільство – це сучасна стадія розвитку людства, в якій головним ресурсом виступає інформація, а провідну роль у всіх сферах життя відіграють знання, інформаційні технології та засоби комунікації [1]. Активний розвиток інформаційного суспільства базується на побудові орієнтованого на інтереси людей, доступного для всіх ресурсу, у якому кожен міг би знайти інформацію та мати вільний доступ з будь-якої точки. Таким ресурсом стала мережа інтернет від створення глобального сервісу World Wide Web (WWW) із системою вебсайтів, пов'язаних гіпертекстом, що є необхідною умовою формування єдиного інформаційного простору. У цьому контексті вебсайти стали основним інструментом поширення інформації. Вебсайт – це сукупність взаємопов'язаних вебсторінок, що розміщені на сервері в мережі інтернет і мають спільну тематику, структуру та адресу. Вони забезпечують користувачам постійний доступ до інформаційних ресурсів у будь-який час і з будь-якої точки світу, що робить їх надзвичайно ефективними засобами комунікації [2].

На початку 90-х років XX ст. всесвітня павутина складалась лише з вебсайтів. По суті, це були просто сховища інформації, що містили статичні документи. Потік інформації був одностороннім, від сервера до браузера. У процесі розвитку інтернет поступово перетворився в інструмент, що сприяє ефективній діяльності кожного учасника інформаційного процесу, економічних та соціальних відносин, особистого розвитку. Сьогодні прогрес у кожній зі сфер діяльності людини визначається тим, наскільки ефективно використовуються інформаційні технології. Особливо вражаюче вебтехнології змінили спосіб взаємодії бізнесу зі споживачами та підприємцями, спростивши процес

купівлі-продажу товарів та послуг [3], а також способи і засоби знаходження та використання величезних потоків інформації.

Вебсайт перетворився на потужний бізнес-інструмент, який надає користувачам доступ до інформації, забезпечує зворотній зв'язок з ними та приносить прибуток підприємству. Він дає можливість користувачеві знайти потрібну інформацію, здійснює миттєвий доступ до таких сервісів, як пошукові системи, електронна торгівля, електронна пошта, комерційна і соціальна реклама, розсилка повідомлень, інтернет-аукціони тощо [3].

На сьогоднішній день вебсайт є невід'ємною частиною супроводу будь-якої професії, будь-якої сфери діяльності. Підприємство, яке має вебресурс, має більшу кількість клієнтів, потужний засіб популяризації своєї діяльності, а отже й суттєві переваги серед конкурентів. Вебсайт дає можливість партнерам та клієнтам більше дізнатися про діяльність підприємства та скласти уявлення про нього. Наявність такого ресурсу в інтернеті означає, що бізнес розвивається, доступний з будь-якої точки та у будь-який час [2].

Щоб точніше зрозуміти, для чого потрібен сайт підприємству, варто розглянути основні функції, які він може виконувати:

1) інформаційна – представлення нової інформації з деякої тематики, області чи проблеми. Клієнти та партнери будуть проінформовані про оновлення або зміни, сучасні тренди, зможуть висловити свою думку при наявності інтерактивних елементів, зареєструватися і т.ін. [1];

2) маркетингова – просування товарів чи послуг, збільшення попиту на них, статистична інформація, огляди тощо;

3) іміджева – представлення підприємства, створення образу, популяризація діяльності.

За допомогою вебсайту можна швидко отримати доступ до потрібної інформації. З кожним днем з'являються нові шляхи для стимулювання продажів з використанням інтернету, що дає можливість підвищити взаємодію всіх учасників комунікаційного процесу, знаходити необхідні обсяги інформації про наявних та потенційних споживачів-користувачів [3].

Вебдодатки створені для виконання практично будь-якої корисної функції, яку можна реалізувати в мережі інтернет, наприклад:

- покупки через інтернет магазини та дошки оголошень;
- соціальні мережі;
- банківські справи;
- пошук в інтернеті;
- аукціони й біржові операції;
- інтерактивна інформація і т. ін.

Відомі аналітичні компанії, такі як Statista, Data.AI, Similarweb, Semrush здійснюють системний моніторинг та публікують регулярні звіти про популярність інформаційних технологій. Серед них найбільш повним, інтегрованим джерелом вважають щорічні огляди Digital [4]. Цей документ пропонує огляди і для українського сегменту аудиторії. Навіть під час такого потужного виклику, яким є військова агресія на території України, запровадження військового стану, розрив багатьох економічних та інформаційних зв'язків, періодична недоступність мережевих послуг, саме діяльність через інтернет демонструє стійкість, відновлення та зростання. За даними звіту Digital 2025 [5], станом на початок 2021 р. кількість українських інтернет-користувачів була 29,47 млн, що становило 67,6 % населення країни – рівень проникнення (рис. 1.1).

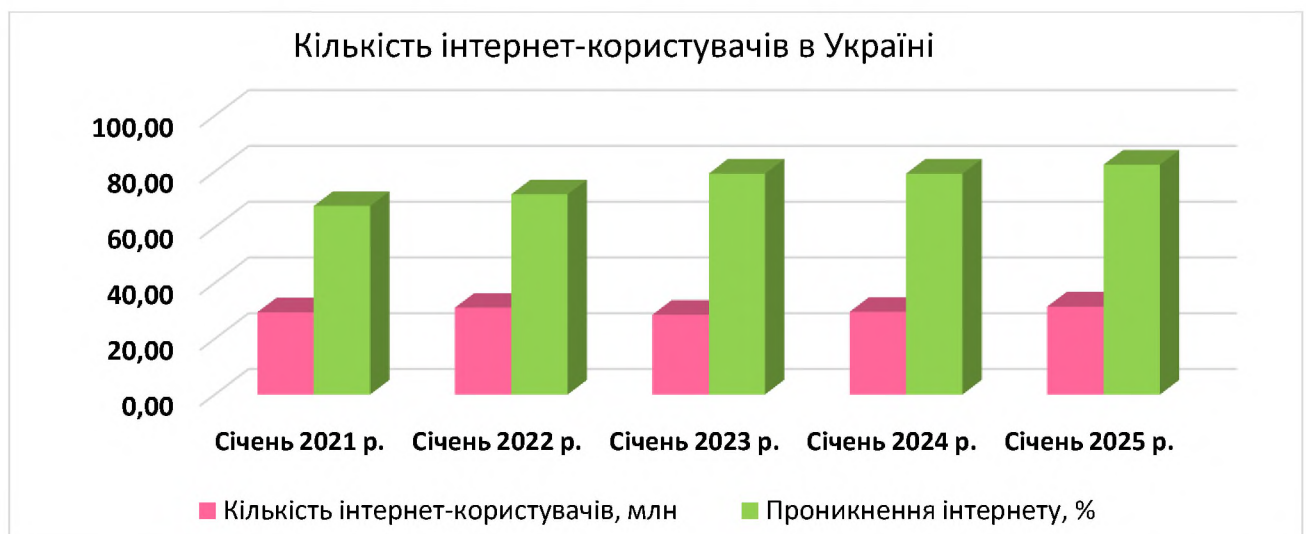


Рисунок 1.1 – Динаміка росту кількості користувачів інтернету в Україні

На початок 2022 р. кількість збільшилася до 31,1 млн, а рівень проникнення інтернету зріс до 71,8 %. З початком війни ці показники зазнали незначних змін, оскільки інтернет став іще більш потужним та затребуваним джерелом інформації та засобом комунікацій. У 2025 р. ця цифра зросла до 31,5 млн (проникнення склало 82,4%).

Отже, найбільш очевидно, щоб отримати більшу кількість нових потенційних клієнтів, будь-якому підприємству, підприємцю, одиниці суспільної діяльності необхідно використовувати ресурси інтернету.

Маркетинг в інтернеті – практика використання усіх аспектів традиційного маркетингу в мережі інтернет, з метою продажу товару чи послуги і управління взаємовідносинами з ним [3]. Застосування інтернет-маркетингу дає підприємствам і бізнесу низку переваг, які допомагають збільшити конкурентоспроможність та імідж і підвищити економічні показники, а отже, примножити прибуток та заощадити кошти.

Імідж підприємства, представлений у грамотності, дизайні, послугах, ціні, стилі та популярності бренду. Він виступає в якості одного з інструментів стратегічних цілей підприємства, що стосуються його діяльності. Імідж є інструментом взаємодії з користувачем через позитивний образ, допомагає досягати цілі у майбутньому.

Сьогодні використання вебтехнологій досить актуальне, адже завдяки їм створюється зручне для користувача інтернет-середовище у вигляді різноманітних порталів, де питання комунікації ґрунтуються на загально визначених стандартах і протоколах. Технології здатні змінювати форми ведення бізнесу і взаємодію між підприємством, наприклад, контроль запасів товарів, електронні закупки та ін. Це дає можливість компаніям отримувати більші прибутки через оперативний обмін інформацією із постачальниками і споживачами та ефективного управління [6].

Сучасний рівень розвитку інформаційної технології характеризується наявністю потужної комп'ютерної техніки, програмного забезпечення, розвинених комунікацій, використання вебтехнологій. Кожного дня

інформаційне суспільство отримує та переробляє інформацію за допомогою комп'ютерної техніки та новітніх технологій [7]. Для підтримки такого способу діяльності необхідним елементом стає наявність сучасного вебсайту [7].

Отже, застосування інтернет-технологій у сфер виробництва, освіти та інших стає одним із найбільш актуальних напрямів просування товарів і послуг та підтримки комунікаційного зв'язку між партнерами та споживачами. Вони сприяють структуруванню та доставці інформації, позитивно впливають на імідж підприємств і організацій.

1.2 Розвиток базових вебтехнологій від початку до сучасних версій

У 1980 р. всесвітньо відомий Тім Бернес-Лі – колишній фізик, який був співробітником CERN, запропонував і прототипував систему INQUIRE, яка повинна була спростити користування документами для дослідників CERN. А вже у 1989 р. запропонував на базі мережі Internet впровадити проєкт розподіленої гіпертекстової системи документів, яку назвав World Wide Web – так звану Всесвітню павутину. На початку ідея полягала в тому, щоб за допомогою гіпертекстової навігаційної системи об'єднати всю безліч інформаційних ресурсів CERN у єдину систему. Вже наприкінці 1990 р. була розроблена мова HTML і написаний браузер та серверне програмне забезпечення для запропонованої системи. Технологія була досить вдалою і дала поштовх до розвитку найпопулярнішої сьогодні глобальної інформаційної системи [8, 9].

Успіх технології WWW визначається двома основними факторами: простотою й використанням протоколів міжмережевого обміну сімейства TCP/IP. Мережна топологія завоювала популярність, оскільки дала користувачам простий інтерфейс для доступу до мережних ресурсів. В результаті збільшились кількість користувачів та з'явилися комерційні структури.

Одним із компонентів створення технології гіпертекстової системи стала мова гіпертекстової розмітки. У 1991 р. було опубліковано в інтернеті перший

загальнодоступний опис технології розмітки HTML, відомий як документ «HTML-теги», який вміщував опис 20 елементів первісної, досить простої схеми розмітки HTML [8]. Мова розмітки HTML дозволяє визначити структуру електронного документа з поліграфічним рівнем оформлення. Результуючий документ може містити різноманітні елементи: текст, графіку, аудіо і відео фрагменти. Технологія HTML включає розвинені засоби для визначення кількох рівнів заголовків, шрифтових виділень, різних груп об'єктів та багато інших можливостей. За основу був звичайний текстовий документ, який легко було створити у будь-якому текстовому редакторі [10]. Згодом такі редактори почали створювати спеціально для написання коду.

Перша версія HTML 1.0 була спрямована на подання мови гіпертекстової розмітки як рекомендацій. Друга версія HTML 2.0 фіксувала практику використання його конструкцій. Версія HTML ++ представляла нові можливості, розширюючи набір тегів HTML у бік відображення наукової інформації й таблиць, а також поліпшення стилю компоновання зображень і тексту. Версія HTML 3.2 змогла впорядкувати всі нововведення й погодити їх з існуючою практикою. HTML 3.2 дозволяє реалізувати використання таблиць, виконання кодів мови JavaScript, обтікання графіки текстом, а також відображення верхніх і нижніх індексів [11].

Подальшу роботу по створенню специфікацій HTML взяла на себе організація World Wide Web Consortium (W3C), у задачі якої входило створення специфікацій, які відображатимуть сучасний розвиток можливостей мови із пропозиціями від компаній-розробників браузерів.

Таким чином, у 1995 р. з'явилася специфікація HTML 2.0, призначена для формалізації практики використання мови [10]. Потреби розвитку вебтехнологій та дотримання певних специфікацій потребували певних угод між розробниками цих технологій. У 2019 році був підписаний визначний документ – Memorandum of Understanding («MOU») – Меморандум про розуміння між консорціумом W3C та спільнотою WHATWG. WHATWG – спільнота фахівців з HTML і API, необхідних для веб-додатків, що займаються розвитком цих технологій і

розробкою стандартів для них. WHATWG була заснована в 2004 році співробітниками компаній-виробників браузерів: Apple, Mozilla Foundation і Opera Software. Сторони оприлюднили домовленість і зобов'язання її дотримуватися в кількох важливих аспектах подальшого розвитку веб.

Цей Меморандум про взаєморозуміння описує процес співпраці для розробки специфікацій HTML і DOM, опублікованих (у минулому чи майбутньому) як W3C, так і WHATWG, де специфікації включають такі, які містяться у версіях WHATWG HTML і DOM, але були опубліковані окремо на сайті W3C. Таким чином, жодна зі сторін-підписантів не може стати монополістом вебтехнологій, так само як і порушувати стандарти [12].

У грудні 1997 р. вийшов офіційний варіант специфікацій HTML 4.0. У цій специфікації основною ідеєю стало відокремлення опису структури документа від опису його представлення на екрані монітору. Це дозволило скоротити витрати на підтримку широкого спектру платформ, середовищ, та спростити виправлення помилок у документі.

Створення вебсторінки, в загальних рисах, полягає у написанні коду за допомогою технологій HTML, CSS, на мовах програмування JavaScript, PHP та інших, і відображенні його у браузері. HTML – це мова розмітки гіпертексту, яка формує блокові елементи сторінки і дозволяє додати контент на неї. Створення HTML-документа починається тегом <HTML>, який є ключовим елементом мови розмітки гіпертексту. Всі теги розпочинаються символом «<» і закінчуються символом «>» та можуть мати атрибути, які, в свою чергу, можуть приймати певні значення. На даний час існує майже 140 HTML тегів, хоча деякі з них уже застарілі та не підтримуються через оновлення браузерів. На сьогоднішній день існує сучасна версія мови гіпертекстової розмітки – HTML5, яка має ряд нових можливостей. Вважається, що HTML5 не є продовженням мови HTML, а нова програмна платформа, для створення вебресурсів. Головною відмінністю від HTML є те, що вона підтримує аудіо та відео, які являються складовою її частиною. Можна виділити основні переваги HTML5 для розробника:

- 1) дозволяє постійно обробляти помилки, що допомагає економити час та кошти розробника;
- 2) для спрощення розуміння коду введені покращення у семантичні ролі важливих елементів, наприклад `<section>`, `<nav>`, `<header>` і т. ін. – нові елементи, які за потреби замінюють більшість `<div>` елементів;
- 3) дозволяє контролювати продуктивність сайту, без застосування JavaScript, або Flash розширень;
- 4) код краще підтримується на мобільних пристроях;
- 5) застосування нового елементу `canvas`, для створення графічних зображень;
- 6) застосування тегу `<menu>`, для кращої інтерактивності на сторінках сайту;
- 7) застосування атрибуту `data`, для зберігання додаткової інформації про різні елементи та включення користувацьких даних, для створення ефективних сторінок сайту;
- 8) підтримка локального сховища даних за допомогою об'єкта `LocalStorage`, який являється частиною глобального простору імен і при використанні скриптів може бути доступним із будь-якого місця [13].

Доповненням для HTML стали каскадні таблиці стилів (Cascading Style Sheets, CSS), за допомогою яких задають спеціальні стилі і визначають, як буде відображатися контент і в цілому зовнішній вигляд сторінки. CSS-правила складаються з селектора і блоку оголошень, всередині якого може знаходитися декілька CSS-властивостей та їх значень, розділених крапкою з комою. Третя версія каскадних таблиць стилів CSS3 дозволяє створювати анімаційні елементи без використання JavaScript. Головна відмінність між CSS та CSS3 полягає у тому, що CSS3 має модулі та підтримує адаптивний дизайн. CSS3 пропонує інструменти для зміни розмірів комірок, які дозволяють отримати правильний розмір будь-якого елементу, різноманітну 3D анімацію, градієнт і кольори RGBA, HSL, HSLA, заокруглення кутів зображень та текстових полів, багатоколонні текстові блоки [14-15].

JavaScript – мова програмування, яка дозволяє створювати динамічний контент, управляти мультимедіа тощо. Коли сторінка завантажується у браузері, то запускається код HTML і CSS після чого виконується JavaScript. Програми на цій мові можуть вбудовуватися у HTML і виконуватися автоматично при завантаженні сторінки у браузері. JavaScript у браузері має доступ до всього, що пов’язано із маніпулюванням вебсторінками, взаємодією з користувачем та вебсервером. Ця мова має повну інтеграцію з HTML і CSS, підтримується усіма основними браузерами та увімкнена за замовчуванням, що є найбільшими її перевагами і особливостями [16].

Через браузер JavaScript може реалізувати такі завдання [16]:

- додавати новий код на сторінку, змінювати її вміст, модифікувати стилі;
- реагувати на дії користувача (клацання миші, наведення миші, натискання кнопок, тощо);
- відправляти мережеві запити на віддалені сервери, зберігати і завантажувати файли;
- задавати питання користувачеві, показувати повідомлення;
- запам’ятовувати дані на стороні клієнта;
- здійснювати елементарну обробку даних.

Шлях від HTML коду у файлі до оформленої сторінки у браузері називається критичним шляхом рейдерингу, його поділяють на два етапи: аналіз коду і побудова дерева рейдера та відтворення сторінки [17].

Дерево рейдера, або дерево візуалізації – це структура, яка складається із HTML-елементів, які будуть відображатися на сторінці і пов’язані з ними стилі. Таке дерево складається з двох компонентів: об’єктної моделі CSS та DOM [18].

Об’єктна структура сайту DOM – це об’єктне уявлення вихідного HTML-документа, для різних програм, простіше кажучи – дерево вузлів. Воно має один корінь, який розгалужується на безліч дочірніх гілок, кожна з яких може розгалужуватися сама і закінчуватися «листям» [18]. Корінь – це HTML-елемент, а гілки – вкладені елементи. Дві частини об’єктної структури є обов’язковими – елементи `<head>` (головна, керуюча частина) і `<body>` - тіло вебсайту (рис. 1.2).

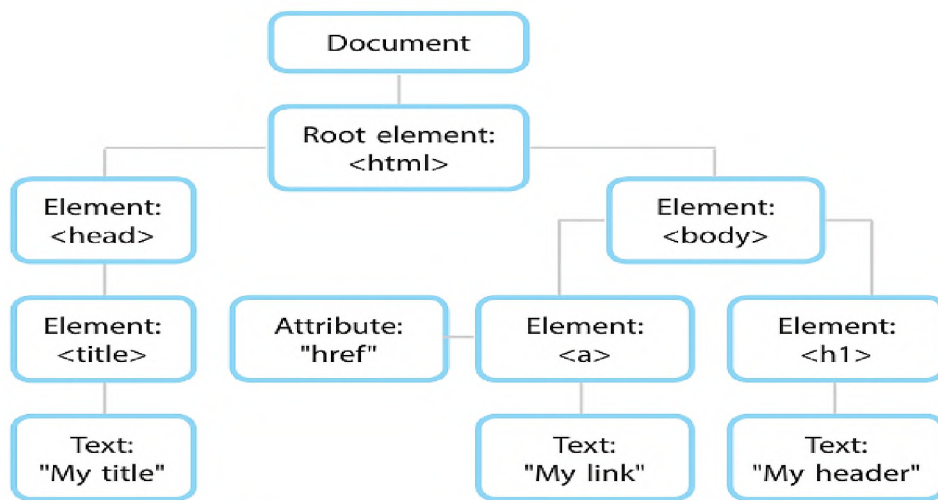


Рисунок 1.2 – Об’єктна структура документу HTML DOM

Тобто, починається все із загального Document, далі в нього вкладається елемент `<HTML>`, наступними сліднують `<HEAD>` і `<BODY>`, а також всі наступні елементи у правильній послідовності. Саме DOM дозволяє отримати доступ до елементів сторінки за допомогою JavaScript.

Використання мови програмування JavaScript є дуже актуальним і тому можна виділити переваги її застосування [19]:

- 1) розповсюджена мова із великою кількістю ресурсів;
- 2) відкривається у кожному інтернет-браузері;
- 3) послідовна розмітка;
- 4) офіційні вебстандарти;
- 5) інтегрується із серверними мовами PHP, Node.js.

Сьогодні навіть програміст-початківець може створити потужний веб-додаток за короткий проміжок часу, користуючись спеціалізованою платформою для розробки вебдодатків та спеціальними інструментами (фреймворками), які надають вже готові компоненти оформлення загального функціоналу, наприклад, форму автентифікації, шаблони сторінок, дошки оголошень, а також реалізують інтеграцію із загальними компонентами інфраструктури. Такі програмні засоби належать до широкого спектру видів програмного забезпечення, але їхні задачі і властивості підпадають під визначення інтегрованих середовищ розробки і програмування.

1.3 Розширення можливостей застосування JavaScript на основі фреймворків і бібліотек

Для JavaScript розробки також є багато корисних технологій та інструментів, наприклад фреймворки, такі як React, Angular, Vue.js, які дозволяють створювати складні вебдодатки швидко та ефективно.

Найпопулярнішими фреймворками та бібліотеками є: React, Angular, Vue.js, jQuery. Згідно аналітичних звітів компанії W3Tech [20], 18.6% вебсайтів не використовують жодну з бібліотек JavaScript, які були в зоні опитування, jQuery використовується на 73,8 % усіх вебсайтів (додаток А, верхня смуга), що складає 90,6 % частки ринку бібліотек JavaScript (нижня смуга діаграми).

Фреймворки (табл. 1.1) являють собою чітку структуру додатку і реалізуються з використанням так званих патернів проектування. Найбільш широко використовувані патерни це: MVC (Model-View-Controller), MVP (Model-View-Persistor), MVVW (Model-View-ViewModel). Додатків на фреймворках доволі багато, і цей сегмент тільки набирає темп [21].

Таблиця 1.1 – Порівняння фреймворків мови програмування JavaScript

Назва	Переваги	Недоліки
React	Ефективність, гнучкість, зручна система багаторазових компонентів, підтримка спільноти, просте налагодження	Погана документація, важкість пошукової оптимізації
Angular	Великий вибір функцій, ймовірність збоїв дуже мала, отримання даних напряму, підтримка Google та потужна екосистема	Високий поріг входу, важкість міграції за старої версії на нову
Vue.js	Простота, легкість інтеграції, дружній інтерфейс, хороша документація	Відсутність підтримки від великих проєктів, відсутність плагінів
jQuery	Простота, прискорення роботи веброзробника, оптимізація під застарілі браузер, безліч готових рішень	Застарілий API, нема рівня даних, не підходить для комплексних задач

React – це бібліотека JavaScript для створення інтерфейсів користувача. Вона дозволяє розробникам створювати вебсайти та вебдодатки, які можуть

швидко та ефективно оновлюватися без перезавантаження сторінки. React був розроблений компанією Facebook і випущений в 2013 році.

React базується на концепції компонентів, що дозволяє розробникам створювати окремі блоки коду, які можуть бути використані повторно в різних частинах вебдодатку. Це робить код більш організованим та легким для зберігання та розуміння.

React також використовує віртуальну DOM (Document Object Model), що дозволяє зменшити кількість змін, які необхідно робити безпосередньо в реальному DOM, що забезпечує більш швидку та ефективну роботу вебдодатку і вона має кілька переваг порівняно з іншими бібліотеками:

1. Єдина «парасолька»: React є лише бібліотекою для відображення інтерфейсу, а не повноцінним фреймворком. Це означає, що можливо використовувати React разом з будь-яким іншим інструментом або бібліотекою, що надає більшу гнучкість та контроль над вашим проектом.

2. Висока продуктивність: React використовує віртуальну DOM (Document Object Model), що дозволяє реалізувати швидкі та ефективні переробки елементів візуального інтерфейсу. Це дозволяє досягнути високої продуктивності вебдодатків, навіть при великій кількості даних.

3. Модульність: React дозволяє розбити ваш інтерфейс на незалежні компоненти, що підвищує його модульність. Це означає, що можливо повторно використовувати код компонентів, що дозволяє зменшити час розробки та покращити підтримку вашого коду.

4. Легкість навчання: React має просту та логічну структуру, що дозволяє легко навчитися розробці з його використанням. Більшість розробників, які знайомі з JavaScript, можуть швидко вивчити React.

5. Гнучкість: React дозволяє змінювати та розширювати його функціонал за допомогою сторонніх бібліотек та плагінів. Це дозволяє збільшити функціональність вашого проекту та забезпечити більшу гнучкість в розробці.

6. Велике співтовариство: React має велику та активну спільноту розробників, яка забезпечує підтримку, допомогу та постійне вдосконалення

бібліотеки. Це означає, що можливо швидко знайти відповіді на свої запитання та знайти відповідні рішення для вашого проекту.

7. Розробка мобільних додатків: React Native - це фреймворк для розробки мобільних додатків з використанням React. Він дозволяє розробляти нативні додатки для iOS та Android, використовуючи звичний для розробників React підхід до розробки інтерфейсу. Це забезпечує зменшення часу розробки та спільний код для різних платформ.

Також варто відзначити, що React є дуже популярною технологією, тому використання його в проекті може забезпечити більшу кількість вакансій та більшу кількість розробників, знайомих з цією технологією [22].

Vue.js – це прогресивний фреймворк для створення користувацьких інтерфейсів вебдодатків. Він дозволяє розробляти високоякісні вебдодатки з високою продуктивністю та ефективністю. Основні переваги Vue.js порівняно з іншими фреймворками можна показати за критеріями [23]:

1. Модульність: Vue.js дозволяє розбити ваш інтерфейс на незалежні компоненти, що підвищує його модульність. Це означає, що можливо повторно використовувати код компонентів, що дозволяє зменшити час розробки та покращити підтримку вашого коду.

2. Легкість вивчення та використання: Vue.js має просту та зрозумілу структуру, що дозволяє легко вивчити його використання. Також він має документацію, яка дозволяє швидко знайти рішення для більшості проблем.

3. Простота інтеграції: Vue.js можна легко інтегрувати з будь-яким іншим інструментом або бібліотекою, що забезпечує більшу гнучкість та контроль над вашим проектом.

4. Висока продуктивність: Vue.js використовує віртуальну DOM, що дозволяє забезпечити високу продуктивність та ефективність роботи вебдодатків.

5. Активна спільнота розробників: Vue.js має велику та активну спільноту розробників, яка забезпечує підтримку, допомогу та постійне вдосконалення фреймворку.

Angular – це повноцінний фреймворк для розробки вебдодатків, розроблений і підтримуваний компанією Google. Він базується на мові програмування TypeScript і має вбудовану підтримку компонентної архітектури.

Переваги Angular порівняно з іншими фреймворками:

1. Широкі можливості: Angular має вбудовану підтримку багатьох функцій, таких як маршрутизація, форми, аутентифікація, перевірка стану і т. ін. Це дозволяє створювати повноцінні вебдодатки без додаткових зусиль.

2. Строга типізація: TypeScript забезпечує строгу типізацію, що допомагає зменшити кількість помилок в ході розробки та полегшує роботу з кодом.

3. Компонентна архітектура: Angular побудований на компонентній архітектурі, що дозволяє розбити додаток на незалежні компоненти, що підвищує модульність і забезпечує можливість повторного використання коду.

4. Оптимізація продуктивності: Angular використовує віртуальну DOM, що дозволяє досягнути високої продуктивності вебдодатків, навіть при великій кількості даних.

5. Підтримка компанією Google: Angular є фреймворком, який розробляється і підтримується компанією Google, що забезпечує його стабільність, підтримку та постійний розвиток.

6. Можливості тестування: Angular має вбудовану підтримку для тестування, що дозволяє легко виявляти та виправляти помилки в ході розробки вебдодатку.

7. Широке співтовариство: Angular має велике та активне співтовариство, що забезпечує наявність безлічі різноманітних додатків, бібліотек, пакетів та рішень для різних задач.

Загалом, Angular є потужним фреймворком з багатьма перевагами порівняно з іншими. Його вбудовані можливості, строга типізація, компонентна архітектура, оптимізація продуктивності, підтримка Google, можливості тестування та активне співтовариство роблять Angular популярним вибором для розробників вебдодатків. Однак, використання Angular потребує певного рівня володіння TypeScript та інших технологій.

Також для покращення розробки JavaScript можна використовувати бібліотеки, такі як jQuery, Lodash, Moment.js та інші, які дозволяють працювати з DOM, маніпулювати даними та працювати зі строками та датами.

Для організації та керування структурою вебдодатку можна використовувати різні інструменти, такі як Webpack, Gulp, Grunt, Parcel, які дозволяють автоматизувати процес збірки та оптимізації вебдодатків [24].

Окрім того, для розробки вебдодатків можна використовувати різноманітні інструменти тестування, які дозволяють перевіряти працездатність та функціональність додатку, такі як Jest, Mocha, Chai та інші.

JavaScript можна використовувати, наприклад, для створення форми замовлення компонентів, перевірки достовірності даних, введених користувачем, та передачі даних на сервер. Анімація, зокрема анімовані графіки, діаграми, графіки тощо, може бути створена за допомогою JavaScript. Це може допомогти у візуалізації даних і підвищити привабливість вебсайту для користувачів.

РОЗДІЛ 2

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИБІР МЕТОДОЛОГІЇ СТВОРЕННЯ ВЕБСАЙТУ НА ЕКОЛОГІЧНУ ТЕМАТИКУ

2.1 Особливості інформаційної підтримки діяльності, направленої на популяризацію екологічності діяльності людини та збереження довкілля

У процесі виконання завдань кваліфікаційної роботи заплановано проектування і реалізацію інформаційного вебсайту на екологічну тематику. При цьому будуть вирішуватися кілька паралельних задач: створення вебсайту як програмно-інформаційного продукту з актуальної тематики, а також закріплення практичних навичок із застосування властивостей самих вебтехнологій, розвинення дизайнерських навичок, досягнення рівня технологічності рішення, планування робіт і т. ін.

Щодо тематики, то актуальність пов'язана з тим, що стан навколишнього середовища викликає дедалі більшу стурбованість у світі, зокрема й в Україні. Забруднення повітря, води, ґрунтів, масове вирубування лісів, зміна клімату, накопичення відходів та зникнення біорізноманіття – все це проблеми, які вже сьогодні впливають на якість життя людей і потребують широкого інформування населення та активної участі громади у вирішенні екологічних питань.

Незважаючи на наявність окремих інформаційних джерел, у суспільстві відчувається дефіцит доступної, структурованої та перевіреної екологічної інформації. Більшість ресурсів орієнтовані або на вузьку аудиторію (експертів, науковців), або мають фрагментарний характер і не відповідають запитам пересічного громадянина чи активістської спільноти [25].

Метою створення вебсайту з екологічної тематики є формування онлайн-платформи, яка стане:

- надійним джерелом знань про довкілля;
- інструментом екопросвіти для широкої аудиторії;

– майданчиком для обміну досвідом та об'єднання зусиль громадських організацій, ініціативних груп і активістів.

Щодо цільової аудиторії, то передбачається, що такий вебсайт має бути корисним для:

- звичайних громадян, які хочуть дізнатися більше про екологічну ситуацію, навчитися дбати про природу в побуті;
- освітян, які шукають матеріали для навчального процесу;
- громадських організацій, що працюють у сфері захисту довкілля;
- місцевих ініціативних груп, які прагнуть реалізовувати екопроекти;
- журналістів, які готують екологічні матеріали;
- органів місцевого самоврядування, які шукають приклади вдалих екологічних практик.

Україна на шляху до Євросоюзу вже сьогодні має бути активно залучена у ці просвітницькі інформаційні процеси. Був проведений аналіз значної кількості вебсайтів подібної тематики на теренах українського інтернет-простору, наприклад, рис. 2.1-2.2.

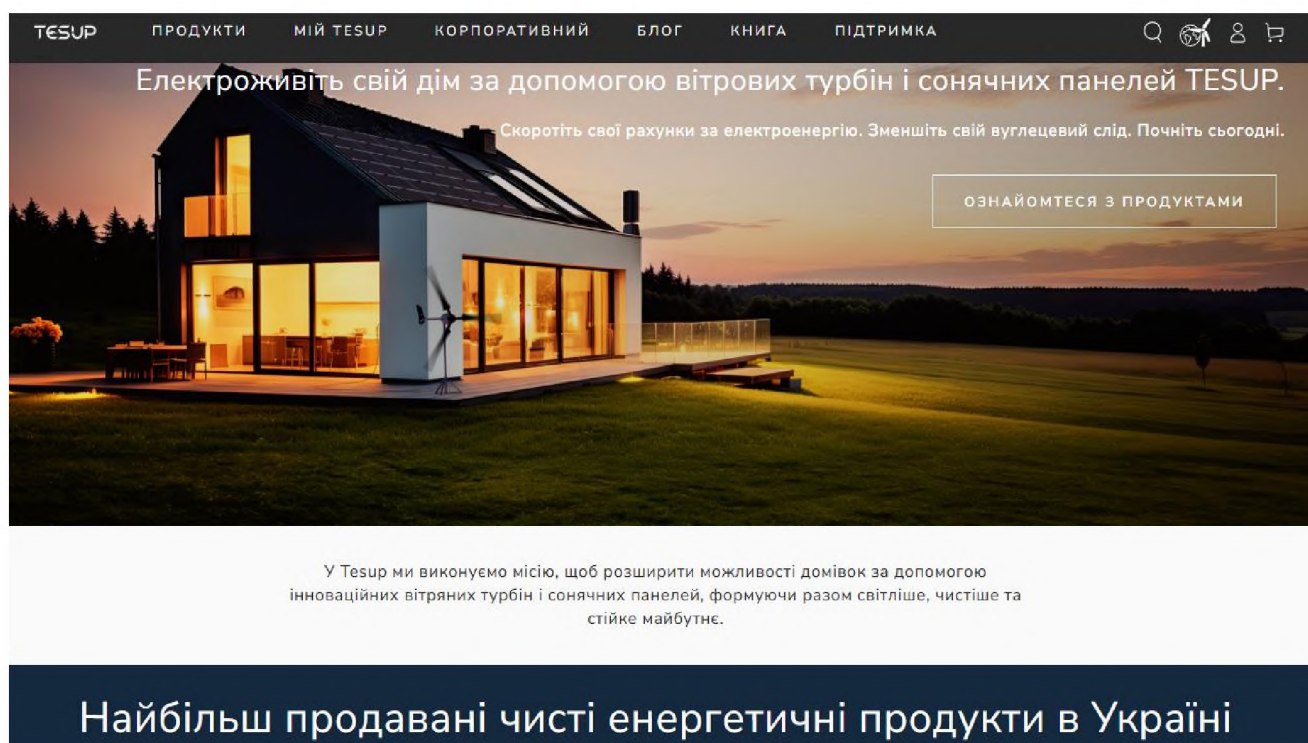


Рисунок 2.1 – Головна сторінка вебсайту про «зелену» енергетику [26]

№ з/п	Екологічні заходи. Ціннісне ставлення до природи Заходи	Дата проведення
1	Догляд за клумбами та зеленими насадженнями, догляд за рослинами живого куточка	Протягом року
2	Проведення екскурсій до лісу, річки. „Зазирни в природу нашого краю”	Вересень
3	День юного натураліста „Юнат – це на все життя”	Вересень
4	Участь в екологічних акціях «Гітач року», «Парад квітів біля школи», «Юний дослідник»	Вересень
5	Виховна година «Ми відповідаємо за тих, кого приручили»	Вересень
6	Оголошення старту екологічних акцій: «Первоцвіт», «Посади дерево», «Чисте джерело», «Не рубай ялинку», «Замість ялинки зимовий букет», «Допоможемо птахам взимку»	Вересень
7	Свято квітів та осінніх композицій: «Квіти – це історія нашого народу»	Вересень
8	Виставка-конкурс на кращу аплікацію з природного матеріалу «Осінній вернісаж»	Жовтень
9	Рейд «Зелені коридори та класи рідної школи»	Жовтень
10	Акція: «Чисте подвір'я – чиста Земля»	Вересень
11	Віртуальний виховний захід „Будьмо друзями природи”	Жовтень
12	Еко-подорож «Віч-на-віч з природою»	Жовтень
13	Трудовий десант: „Чиста школа – чиста планета”	Жовтень
14	Конкурс малюнків, фотографій та літературних творів «Земле –любове моя!»	Жовтень
15	Прийняти участь в операції «Не паліть листя»	Жовтень
16	Тематичні екологічні свята: «Природа – мати-берегиня»; вікторини «Біологічний хокей»; лото «Корисні копалини», «Щасливий випадок»	Жовтень
17	Конкурс на кращу екологічну рекламу «Від відходів – до доходів»	Жовтень
18	Конкурс екологічних плакатів: «Збережемо цей неповторний світ!»	Жовтень
19	Усний журнал „Рідкісні птахи нашої місцевості”	Листопад

Рисунок 2.2 – Фрагмент інформаційної сторінки з переліком екозаходів [27]

Найбільш різноманітні матеріали знайдено на вебсайтах університетів: це матеріали конференцій, навчальні посібники, курси лекцій та інші матеріали навчального змісту. Однак, більшість із них представлені у вигляді електронних текстів, іноді презентацій. Вся ця інформація має достатньо високий інтелектуальний рівень, але не зібрана для зручного ознайомлення для широкого загалу користувачів.

На іншому полюсі – комерційні вебсайти з пропозиціями різноманітного обладнання та технологій для енергозбереження (див. рис. 2.1), охорони довкілля та комерційним пропозиціями. Такі компанії мають гарно оформлені багатосторінкові сайти, які окупаються за рахунок реклами та продажів продукції [26].

Узагальнивши типові типові недоліки існуючих вебсайтів на екологічну тематику, вийшов перелік:

- низька зручність користування (UX/UI): багато ресурсів мають застарілий або перевантажений інтерфейс, що ускладнює пошук інформації;
- недостатня актуальність: сайти не оновлюються регулярно, новини або дані можуть бути застарілими;
- надмірна академічність: інформація подається у складній, малозрозумілій формі для нефаківців;
- відсутність інтерактивності: користувач не може взаємодіяти з сайтом, залишати коментарі, пропонувати власний контент або ініціативи;

- фрагментарність: деякі сайти висвітлюють лише одну тему (наприклад, сортування сміття), і не дають комплексного уявлення про екологічну проблематику;
- недостатнє охоплення місцевого контексту: багато ресурсів зосереджуються на глобальних або національних темах, не приділяючи уваги регіональним проблемам та ініціативам.

Таким чином, інформаційний вебсайт на екологічну тематику є вкрай необхідним для формування екологічно свідомого суспільства [28]. Його створення сприятиме підвищенню рівня екопросвіти, розвитку громадянських ініціатив і кращій взаємодії між державою, громадами та окремими громадянами. Комплексна структура сайту, актуальний і зрозумілий контент, а також орієнтація на залучення користувачів зроблять цей ресурс потужним інструментом у справі збереження довкілля. В наступних підрозділах необхідно остаточно обрати методологію та інструменти розробки.

2.2 Методологія життєвого циклу проєкту з розроблення вебсайту

Предметом інтернет-проєкту є робота з інформацією в мережі. Вони класифікуються за такими ознаками: функціональністю, змістом контенту, технологіями, які використовуються та ін. Кожен такий проєкт має своє завдання і свій життєвий цикл. Життєвий цикл сайту – це період часу його роботи в інтернеті із конкурентоспроможністю. Життєвий цикл проєкту є методологією його реалізації від ініціювання до закриття.

Також можна відокремити основні фактори, які впливають на життєвий цикл сайту та можуть його скорочувати [29]:

- застаріла інформація;
- нові завдання, які виконує сайт;
- невдалий інтерфейс;
- зростання конкуренції;

– застаріле програмне забезпечення.

Життєвий цикл створення сайту відбувається протягом основних етапів:

1. Етап ініціації (задум, визначення цілей);
2. Проектування;
3. Розроблення (верстка);
4. Тестування, оптимізація;
5. Впровадження;
6. Адміністрування сайту.

На початковій стадії визначаються основні цілі та завдання сайту, тобто створюється опис проекту, технічне завдання. Це досить важливий етап, адже грамотно складене технічне завдання (ТЗ) дозволить чітко визначити порядок робіт, терміни виконання, уникнути непорозумінь між замовником та виконавцем та економить час на створення сайту. ТЗ повинно вміщувати групи розробників, обсяг та графік робіт, фінансування і узгоджуватися із замовником [8, 9, 30].

Етап проектування вміщує створення прототипу сайту, для наочного представлення розташування елементів на сторінці і узгодження із замовником. Та на цьому етапі створюють макети сторінок. Дизайн сайту відповідає за візуальне його представлення. Слід розуміти, що дизайн має забезпечувати зручність використання сайту (usability), тому всі його елементи повинні зручно розміщуватися, для того, щоб користувач міг легко знайти їх [8, 30].

Наступний етап – розроблення. При розробці сайту необхідно визначити кількість сторінок та зв'язки між ними. Виділяють ієрархічну, лінійну, ґратчасту та змішану структури. Далі вказано види побудови сайту [9, 30].

Ієрархічна – основна інформація розміщується згори вниз. В основному головна сторінка розміщується зверху, яка не має попередніх і містить посилання на всі інші сторінки. Навігація спрямована згори вниз із посиланнями вгору за ієрархією та навпаки. Цей спосіб найпоширеніший. Така структура дає відвідувачам можливість легкого пошуку інформації на сайті від загальної інформації до більш конкретної (рис. 2.3.) [8].

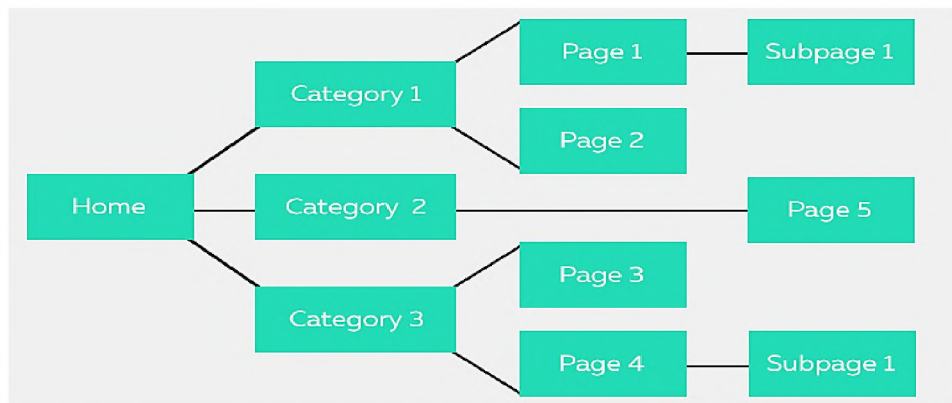


Рисунок 2.3 – Схема ієрархічної (деревоподібної) структури вебсайту

Лінійна, або послідовна, або каскадна структура складається із послідовно розміщених сторінок одна за одною у вигляді, так званого ланцюга. За даним способом користувачі можуть переміщуватися тільки в одному напрямку – послідовно, тому така структура зручна для створення невеликого сайту із декількох сторінок, або для реєстрації користувачів на інтернет-ресурсі. Таке розміщення інформації ефективно для представлення тексту у послідовному порядку, адже все відбувається за конкретним маршрутом [30]. Схематично лінійну структуру подано на рис. 2.4.

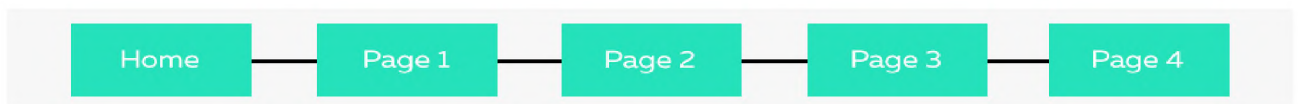


Рисунок 2.4 – Схема лінійної структури сайту

Гратчаста, або таблична, структура полягає у швидкій навігації між вертикальними та горизонтальними сторінками без переходу на проміжні сторінки (рис. 2.5), найчастіше застосовується для створення каталогів.

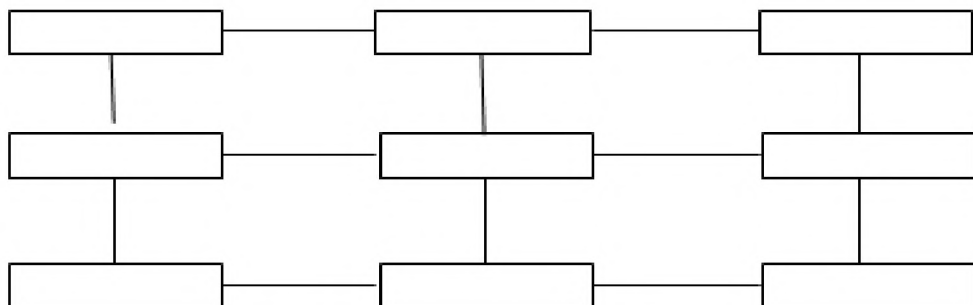


Рисунок 2.5 – Схема гратчастої структури вебсайту

Змішана структура полягає у побудові системи навігації сайту, із поєднанням двох або більше компонентів, а саме ієрархічної та лінійної структур. Така структура доволі складна, адже всі сторінки розміщені на окремих розгалуженнях (рис. 2.6).

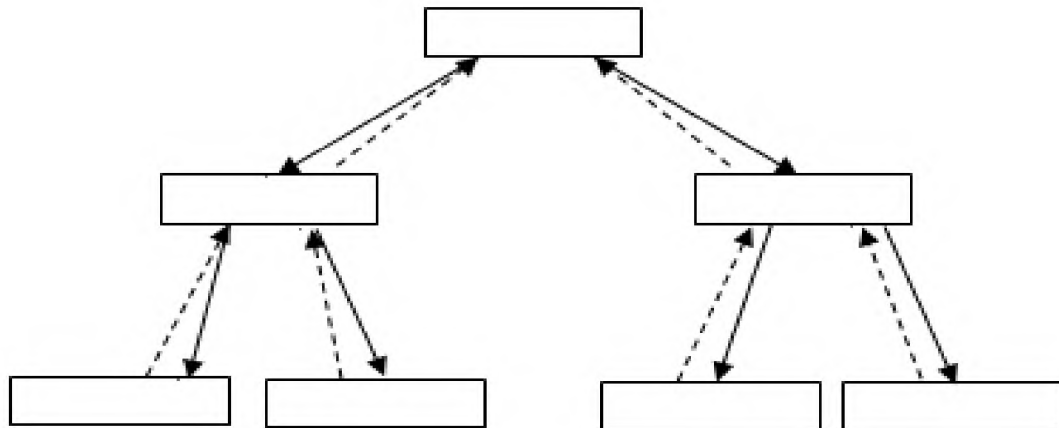


Рисунок 2.6 – Схема змішаної структури сайту

Також на етапі розробки слід підібрати матеріал для розміщення на сторінках сайту та обрати програмні засоби, за допомогою яких буде розроблено сайт. Після чого відбувається наповнення сайту матеріалами та створення гіперпосилань для зручної навігації по сайту. При цьому вся інформація повинна бути унікальною та грамотною із використанням графіки та анімацій [8].

Готовий сайт вимагає тестування для перевірки функціонування та виправлення помилок. На цьому етапі необхідно перевірити роботу всіх гіперпосилань, зручність використання, відображення всіх графічних та анімаційних елементів тощо.

Для цього існує декілька видів тестування вебсайтів [31]:

1. Тестування функціональності;
2. Перевірка зручності використання;
3. Тестування інтерфейсу користувача;
4. Тестування продуктивності;
5. Тестування безпеки.

Функціональне тестування спрямоване на перевірку коректної роботи функціональності сайту, тобто коректність реалізації функціональних вимог. Включає перевірку роботи всіх посилань, форм, різноманітних функцій [31].

Тестування зручності використання спрямоване на дослідження того, наскільки зручно і зрозуміло користуватися сайтом кінцевим клієнтам. Основними характеристиками є: простота навчання, навігація, задоволення користувача, загальний вигляд. Тобто, сайт повинен бути зручним і логічним у використанні, послідовним, головне меню розміщувалося на всіх сторінках [32].

Тестування інтерфейсу користувача спрямоване на перевірку графічного інтерфейсу або його компонентів і включає перевірку інтерфейсу кнопок, формат і шрифт тексту, зображень та інтерфейсу всього сайту. На цьому етапі здійснюється перевірка на сумісність із різними інтернет-браузерами [31].

Тестування продуктивності полягає у дослідженні показників швидкості реакції сайту на різні навантаження. При такому тестуванні перевіряється яка кількість користувачів може взаємодіяти із однією сторінкою, на скільки сайт може обробляти одночасні запити користувачів і проводиться перевірка поведінки сайту при змінах навантаження та інше [31].

Тестування безпеки спрямоване на перевірку здатності сайту протистояти зловмисним спробам отримання доступу до даних без прав доступу. Полягає у пошуку всіх слабких сторін, які можуть призвести до втрати даних.

У результаті проведення всіх тестувань повинен бути створений документ.

Наступний етап – впровадження. На даному етапі необхідно обрати домен сайту, який буде легко запам'ятовуватися, та обрати компанію, яка надає послуги з хостингу. Вже готовий сайт може розміщуватися на: власному сервері, сервері провайдера, на сервері організації з якою співпрацює підприємство, або безкоштовному сервері [33]. Етап впровадження включає:

- вибір доменного імені, його реєстрацію, вибір компанії, що забезпечує хостинг та розміщення сайту в мережі інтернет;
- створення звітів користувачів із результатів експлуатації сайту;
- підтримка сайту виконавцем.

Процес розміщення сайту в інтернеті передбачає надання йому адреси.

Домен – частина адресного простору в мережі інтернет, призначена для ідентифікації комп'ютера або групи комп'ютерів [33]. Доменне ім'я – цифровий вираз, що ідентифікує комп'ютер або абонента у мережі інтернет.

Домени першого рівня – мають визначені імена і поділені на дві групи за тематичними та географічними ознаками. Домени ua, uk, us є географічними доменами України, Великобританії та США.

До тематичних доменів відносять: com – комерційні організації, org – неурядові, edu – освітні, net – домен організацій, що надають мережні послуги та ін. Домени інших рівнів обираються із врахуванням назви підприємства, організації чи теми сайту [34].

Реєстрація доменного імені дозволяє створити не лише бренд, але і дає можливість просування в мережі. Отже, для якісного просування сайту воно повинно легко запам'ятовуватися, бути коротким, або означати сферу діяльності.

На сьогодні існує велика кількість компаній, які надають послуги хостингу, багато з них платні. Але є можливість і безкоштовного розміщення сайту в мережі (з рекламою). За місцем хостинг може бути:

- 1) віртуальний – надається частина місця на диску для розміщення вебсайту [34];
- 2) виділений сервер – сервер для ексклюзивного використання, для великих проєктів із високим ресурсом;
- 3) віртуальний виділений сервер – надається сервер кільком ресурсам із інфраструктурою ідентичною серверу, але із частковою продуктивністю;
- 4) колокація – надання місця в дата центрі провайдера для обладнання клієнта і підключення його до мережі інтернет [34];
- 5) хмарний хостинг – послуга за якої файли користувача зберігаються на багатьох серверах;
- 6) реселлер хостинг – хостинг із послугою перепродажу.

Останній етап створення сайту це його підтримка. Для ефективної роботи сайту необхідно слідкувати за актуальністю контенту на сайті, вносити зміни до

його оформлення. Також на цьому етапі відбувається оптимізація сайту після внесених змін та під пошукові запити користувачів.

У результаті всіх етапів створено проєкт вебсайту, за допомогою якого можна отримати уявлення про очікувані результати, приблизний кошторис, а отже контролювати подальше створення сайту і досягти поставлених цілей.

2.3 Формування стеку технологій з розроблення вебсайту обраної тематики

Перед розробкою вебсайту після визначення його основної ідеї, цільової аудиторії, функціональності слід обрати відповідне програмне забезпечення, на основі якого і в середовищі якого буде створено сайт. Всі методи розробки сайтів можна умовно розділити на дві групи: ручне та автоматизоване [35].

Автоматизований метод полягає у створенні вебсайту за допомогою спеціальних конструкторів або систем керування контентом. Він передбачає поділ структури сайту на дизайн та контент [35].

Конструктори сайтів – це онлайн-системи, що створюють сайт із типового набору модулів і компонентів із розміщенням його у мережі. Часто конструктори сайтів генерують HTML код, який майже не можливо відредагувати і він займає багато місця, а отже сторінки будуть довго завантажуватися. Такий метод підходить для створення простих сайтів, але він не гарантує розробку унікального сайту, а деякі можливості можуть бути обмежені. За користування такими сервісами, як правило, доводиться платити.

Ручне написання сайтів здійснюється у текстових або спеціальних редакторах HTML та CSS кодів із підтримкою використання JavaScript, PHP і т. ін. Текстові редактори поділяють на редактори текстового режиму і віконні редактори. Віконні редактори виглядають більш сучасно і володіють деякими корисними елементами користувацького інтерфейсу. Створення та редагування може здійснюватися у віконних редакторах, наприклад, Sublime Text,

Notepad ++, TextPad тощо. Цей процес досить складний та трудомісткий, адже він вимагає навичок в області вебпрограмування і вебдизайну.

Для реалізації завдань практичної частини кваліфікаційної роботи було обрано саме ручний метод, адже він дає можливість вибору конструкції вебресурсу. В якості середовища проєктування обрано продукт компанії Microsoft Visual Studio Code, або VS Code, який є одним із найпопулярніших інтегрованих середовищ для веброзробки (IDE). Він пропонує широкий спектр функціональних можливостей та розширень, які допомагають веброзробникам підвищити продуктивність та зручність роботи. Для вибору ідеального IDE доречно ознайомитися з думками професійних розробників, даними аналітичних компаній. На рис. 2.7 наведено рейтингову таблицю з 4-х найкращих станом на 2022 р. згідно [36].

Rank	Change	IDE	Share	1-year trend
1		Visual Studio	28.23 %	+0.1 %
2		Visual Studio Code	13.51 %	-0.5 %
3		Eclipse	11.66 %	+0.0 %
4	↑	pyCharm	10.73 %	+2.0 %
5	↓	Android Studio	9.64 %	+0.6 %
6		IntelliJ	7.45 %	+0.7 %

Рисунок 2.7 – TOP IDE index: рейтинг популярності найбільш відомих IDE

Однією з основних переваг VS Code є його інтеграція з багатьма популярними інструментами та технологіями веброзробки. Наприклад, VS Code має підтримку для мов програмування, таких як HTML, CSS, JavaScript, Python та інших. Він також підтримує інтеграцію з Git, що дозволяє веброзробникам легко керувати версіями свого коду та співпрацювати з іншими розробниками. Завдяки інтеграції IDE програмісти можуть мати той самий набір можливостей в одному місці без необхідності постійно перемикати інструменти [37]. Для веброзробників і програмістів існує широкий набір програмних засобів від редакторів коду до написання складних додатків на мовах Java, PHP та ін.

Крім того, VS Code пропонує багато корисних функцій, які спрощують написання коду та зменшують кількість помилок. Наприклад, він має автодоповнення коду, що допомагає веброзробникам швидше та точніше писати код. Також VS Code має можливість відлагодження коду, що дозволяє розробникам швидко виявляти та виправляти помилки в своєму коді.

Середовище має широкий спектр розширень та плагінів, що дозволяє розширити його функціональність та адаптувати його під потреби веб-розробника. Наприклад, можна встановити розширення для роботи з фреймворками, такими як React, Angular або Vue, або для роботи з базами даних та іншими інструментами.

Окремо варто розглянути управління версіями коду у Visual Studio Git&GitHub. Це дуже зручна і корисна опція для тих, хто хотів би переглянути попередні версії коду. За допомогою Git можна відстежувати зміни, що вносяться в код з часом, та виконувати відкат до певних версій [38]. Можливості Git у Visual Studio можуть принести велику користь і новачкам, і професійним розробникам. Серед основних переваг VS Code визначають такі:

1. Багатофункціональність: дозволяє вам працювати з багатьма різними типами файлів, включаючи HTML, CSS, JavaScript та інші. Крім того, він підтримує роботу з більшістю відомих фреймворків та бібліотек веброботи.
2. Керування Git: має вбудовану підтримку для системи контролю версій Git, що дозволяє вам легко виконувати коміти, переглядати відмінності та зливати гілки.
3. Відлагодження: має підтримку відлагодження для багатьох мов програмування, що дозволяє вам знайти та виправити помилки у вашому коді.
4. Розширення та плагіни: має велику кількість розширень та плагінів, що дозволяють налаштувати редактор під свої потреби та розширити його функціональність.

Таким чином, VS Code може значно спростити роботу веброзробника та допомогти йому створювати високоякісний код. Завдяки своїм можливостям та розширенням він є відмінним вибором для розробки сайтів та додатків на веб.

Для наочного представлення майбутнього сайту зручно використовувати наперед створений макет. Макет сайту – це його реалістичний прототип, створений у графічному редакторі. Він має вигляд готового сайту із всіма блоками, зображеннями, кнопками і тому подібне. Єдина відмінність у тому, що макет не має функціонального наповнення, тобто містить тільки шаблони описів. Він потрібен для скорочення процесу створення сайту, адже при наявності макету вебпрограмісту легше розробляти сайт із мінімальними дефектами [39].

Можливо виділити основні переваги наявності макету:

- допомагає скласти наочне уявлення про вигляд майбутнього сайту, побачити, як будуть виглядати елементи на сторінках;
- розробити єдиний дизайн для всіх сторінок сайту;
- продумати наповнення елементів текстом, зображенням, відео тощо;
- задати чіткі границі для роботи над створенням сайту.

Після розгляду всіх переваг можна перейти до розробки макету. Вся робота відбувається поетапно. Починається із створення ТЗ, в якому слід указати цілі сайту, короткий аналіз цільової аудиторії, користувацький сценарій, кількість сторінок та затвердити замовником [39]. Порядок розробки та зміст технічного завдання відповідає ГОСТ 34.601–90.

Кожен інтернет-ресурс, від домашньої сторінки до великого інформаційного порталу, містить декілька тематичних рубрик, сполучених між собою гіперпосиланнями. Якщо тематичні рубрики містять власні підрозділи, кожна з них також має свою стартову сторінку.

Не менш важливими є представлення логічної структури сайтів, типи яких розглянуто вище (див. рис. 2.3-2.6). На рис. 2.8 схематично наведена структура сайту, що включає назви розділів сайту і представляє взаємозв'язок усіх сторінок інтернет-застосунку, забезпечує перехід користувача від загальної інформації (у даному випадку це Головна сторінка) до розгорнутої. Кількість переходів між сторінками, кількість кроків до цільової інформації визначає зручність використання (UI дизайн) та скільки часу користувач проведе на сайті.

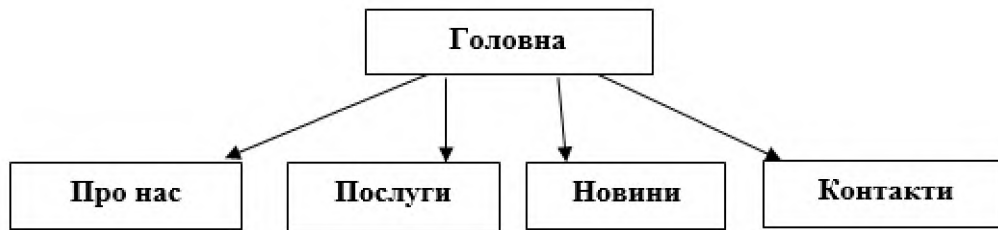


Рисунок 2.8 – Попередня схема логічної структури сайту на екотематику

Зовнішня структура сайту містить в собі розташування видимих блоків на ньому, а саме: шапку, яка має назву ресурсу, логотип та навігаційне меню для переходів між розділами сайту, блоки контенту із текстом, відео, зображенням, службові форми та footer (підвал сайту), в якому розміщені контактні дані та інформація про авторські права (рис. 2.9). Крім таких блоків, на сайті може бути розміщена реклама або посилання на інші інтернет-ресурси. Розробка зовнішньої структури тісно пов'язана з програмною реалізацією сайту.

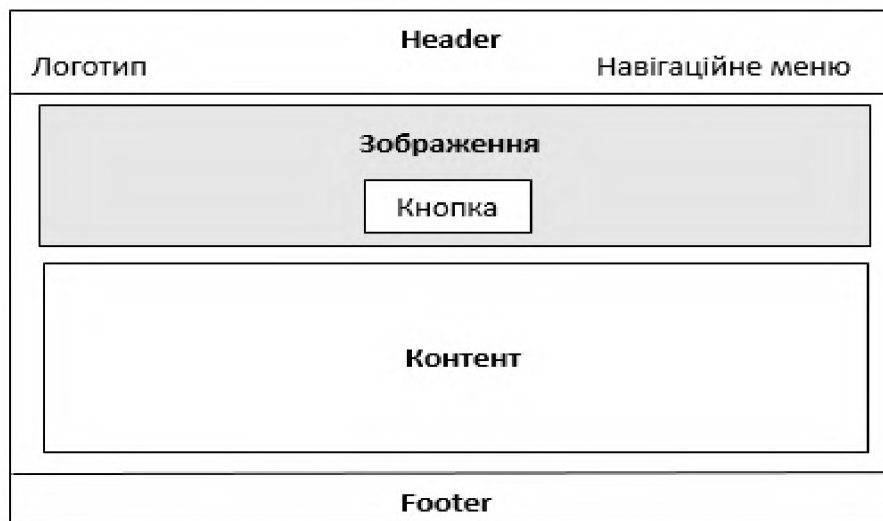


Рисунок 2.9 – Макет зовнішньої структури сайту у формі лендінгу

Макет вебсайту забезпечує зручну основу для подальшої розробки та дизайну вебсторінки. Для розробки макету використовують різноманітні графічні редактори, наприклад Adobe Photoshop, Wix, Figma та інші, за допомогою яких макет створюється досить швидко.

Для макету розроблюваного вебсайту було обрано редактор Figma. У програмі доступні різноманітні інструменти для створення фігур, ліній та інших

елементів дизайну. Крім того, Figma пропонує можливість створювати векторні зображення та використовувати різні типи шрифтів. Окрім роботи з графікою, Figma забезпечує зручну роботу зі структурою документа, дозволяє створювати різні елементи дизайну та організовувати їх у вигляді шарів. Крім того, Figma дозволяє працювати зі створеними документами в команді для роботи над проектами великих масштабів. Фрагмент початку роботи над макетом показано на рис. 2.10.

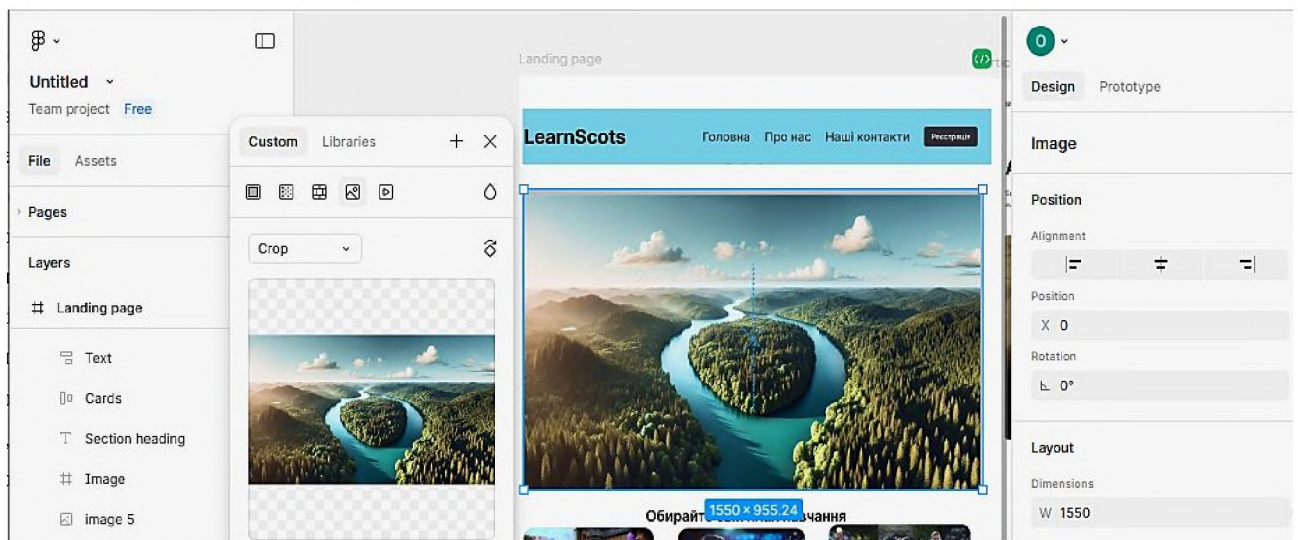


Рисунок 2.10 – Робота над макетом у редакторі Figma

Однією з найважливіших функцій Figma є можливість відстежувати зміни, що вносяться до документу. Це дозволяє ефективно працювати в команді та швидко вирішувати проблеми, пов'язані зі змінами в дизайні. Також варто зазначити, що Figma доступний як для використання в браузері, так і в вигляді додатку для робочого столу. Це дозволяє працювати з програмою в будь-яких умовах та на будь-яких пристроях.

Програму Figma обрано завдяки тому, що вона не лише має інструментарій візуального створення елементів макету вебсторінки, але й спеціальний плагін «Figma to html» [40], який допомагає легко перетворити ваш дизайн Figma на код, не втрачаючи жодної частини дизайну чи кольорів. Це стало можливим за допомогою лише принципів гарного дизайну та керівних ліній, щоб отримати

бажаний результат у вигляді точної копії вашого дизайну у формі коду. Частину головної сторінки, попередньо спроектованої у Figma, показано на рис. 2.11.

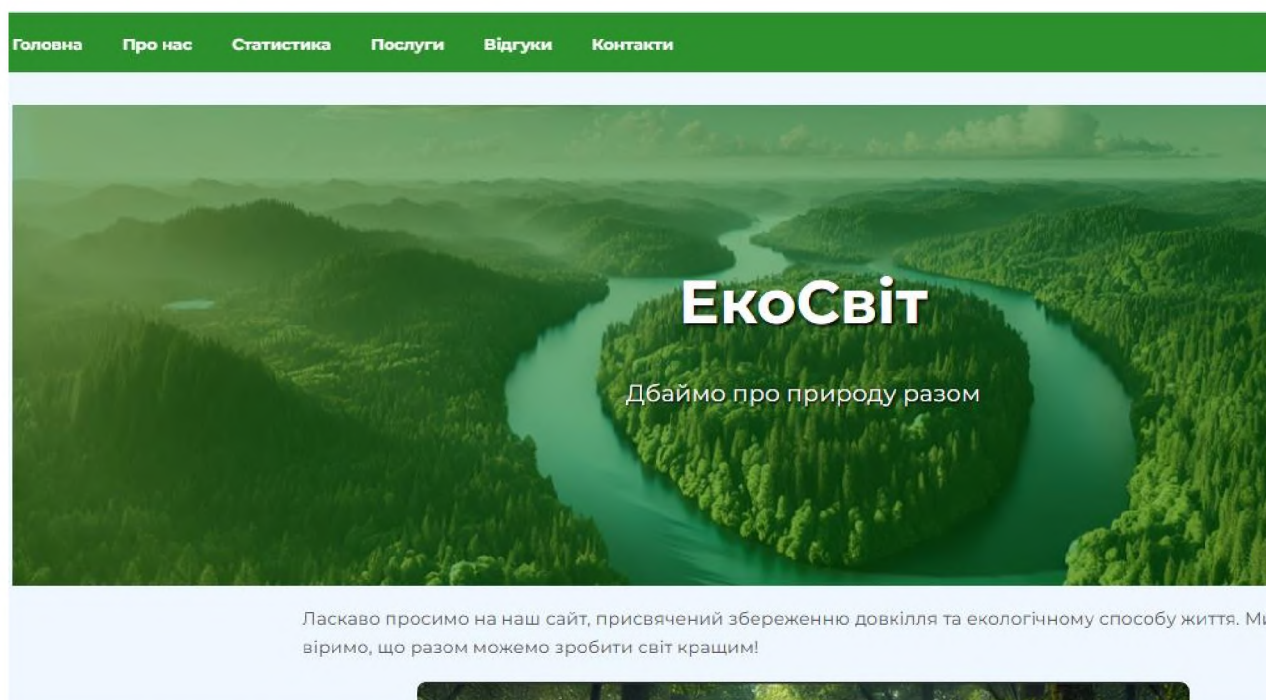


Рисунок 2.11 – Вигляд частини верхньої секції вебсайту, спроектованого у Figma

Загалом, розроблено багатосторінковий сайт з описом унікальних властивостей органічної продукції, вибрані відповідні графічні елементи, розроблені логотипи. У процесі створення макету сайту було враховано його завдання, специфіку предметної області, цільову аудиторію та враження, які він буде справляти на відвідувача.

Провівши аналіз подібних сайтів, можна виділити, що сайти використовують мінімалістичний дизайн без складних і громіздких графічних елементів, простий, зручний для читання шрифт, якісні фото та переважно зелені кольори. Figma – це потужний інструмент для розробки інтерфейсів, який забезпечує зручне та ефективне середовище для роботи з дизайном. Багатофункціональність та можливість працювати в команді роблять цю програму незамінним інструментом для роботи з проєктами будь-якої складності. Наступним етапом є його технічна, програмна реалізація.

РОЗДІЛ 3

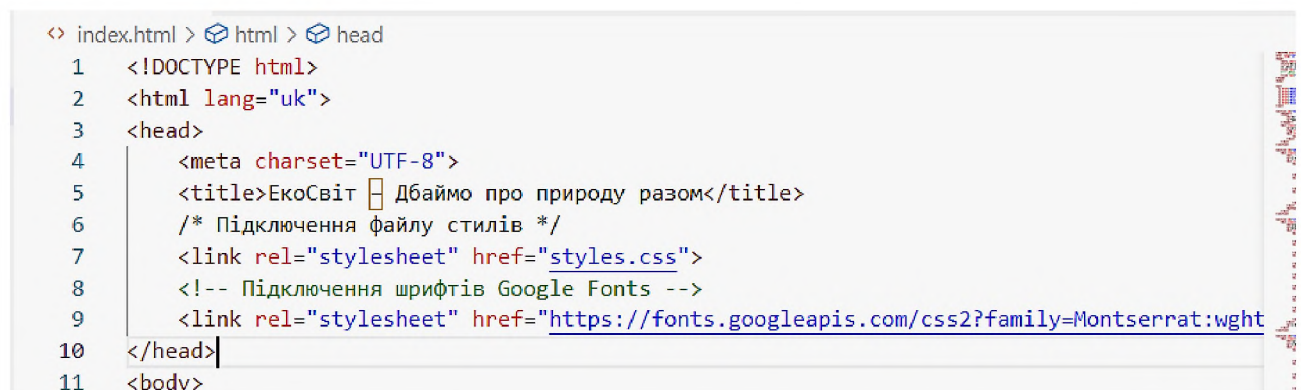
ТЕХНІЧНІ АСПЕКТИ РЕАЛІЗАЦІЇ ПРОЄКТУ РОЗРОБЛЕННЯ ВЕБСАЙТУ НА ЕКОЛОГІЧНУ ТЕМАТИКУ

3.1 Створення HTML-структури та верстання вебсайту

Для розробки вебсайту на екотематику було обрано мову розмітки HTML5 та технологію CSS3, а також мову програмування JavaScript та окремі його бібліотеки. Вибір HTML5 дозволяє створити і структурувати розділи, заголовки, абзаци, посилання та блоки для вебсторінок і додатків, він визначає загальну структуру та організацію контенту, що сприяє покращенню користувацького досвіду та забезпечує більш ефективну роботу з вебсайтом [41]. Як уже зазначалося, сайт розроблено у формі лендінгу. Всі змістовні блоки об'єднані в одному кольоровому оформленні та мають продуману навігацію.

Базова структура вебсторінки, написана в тегах HTML, визначає загальну розмітку та розташування основних елементів, таких як заголовки, текст, зображення, навігаційні меню, футер та інші (іноді називають «скелет» сайту), зазвичай складається з HTML-коду та базового CSS-стилю.

Частина між тегами `<head>...</head>` є головною частиною з управляючими елементами: метатеги, посилання на зовнішні файли стилів, підключення шрифтів, бібліотек за потреби (рис. 3.1)призначений для зберігання службової інформації про всю сторінку.



```

<> index.html > html > head
1  <!DOCTYPE html>
2  <html lang="uk">
3  <head>
4      <meta charset="UTF-8">
5      <title>ЕкоСвіт Дбаймо про природу разом</title>
6      /* Підключення файлу стилів */
7      <link rel="stylesheet" href="styles.css">
8      <!-- Підключення шрифтів Google Fonts -->
9      <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Montserrat:wght
10 </head>
11 <body>
  
```

Рисунок 3.1 – Головна частина вебсайту в index.html в редакторі VSC

Основними частинами, які описуються тегами HTML є:

- навігаційна горизонтальна панель з шістьма пунктами меню, описана семантичним тегом `<nav>` та посиланнями `<a href="#">...</a>`;
- змістові секції у вигляді окремих блоків, які мають кілька рівнів вкладеності, зовнішній блок створений тегом `<section>...</section>`;
- заголовок сайту (в тезі `<header>` розташований головний банер) та кнопка виклику повідомлення;
- зона контенту, яка може включати один або більше блоків тексту та зображень;
- футер сайту, який містить інформацію про копірайт та інші важливі посилання, контактні дані, додаткові посилання.

Повний лістинг коду HTML (198 рядків) показано в додатку Б. Фрагмент коду зі структурою хедера наведено на рис. 3.2: заголовок `<h1>` і текст, потім банер (зображення) з кнопкою, при клацанні на яку спливає повідомлення.

```

22 <section id="home">
23   <header>
24     <h1>ЕкоСвіт</h1>
25     <p>Дбаймо про природу разом</p>
26   </header>
27   <div class="content">
28     <p>Ласкаво просимо на наш сайт, присвячений збереженню довкілля та екологічному способу життя.
29     
30     <div class="button-container">
31       <button onclick="showMessage('Дякуємо, що приєдналися до нас!')">Приєднатися</button>
32     </div>
33   </div>
34 </section>

```

Рисунок 3.2 – Код першої секції загальної інформації

Властивості кожної секції, до якої можна переміститися з відповідного пункту меню, показані в окремому файлі стилів `style.css`. Кожна секція і вкладені блоки мають відповідний стиль в селекторах `id` або `.class`. Для раціонального запису стилів сформована певна система. Наприклад, кожний вкладений блок першого рівня в секції `<section>` описується стилем `<div class="content">` (рис. 3.3). Інші стилі належать різним класам або тековим селекторам (рис. 3.4).

```

36 <section id="about">
37   <div class="content">
38     <h2>Про нас</h2>
39     <p>Ми ☐ команда однодумців, яка прагне підвищити екологічну свідомість суспільства та сприяти зберіг
40     <div class="grid-container">
41       <div class="grid-item">
42         
43         <h3>Наша місія</h3>
44         <p>Зробити екологічний спосіб життя доступним та привабливим для кожного.</p>
45       </div>
46       <div class="grid-item">
47         
48         <h3>Наші цінності</h3>
49         <p>Відповідальність, співпраця, інновації та прозорість.</p>
50       </div>
51       <div class="grid-item">
52         
53         <h3>Наша команда</h3>
54         <p>Професіонали з різних галузей, ☐ об'єднані любов'ю до природи.</p>
55       </div>
56     </div>
57     <button onclick="showMessage('Раді, що ви хочете дізнатися більше!')">Дізнатися більше</button>
58   </div>
59 </section>

```

Рисунок 3.3 – Частитна коду HTML Структура секції «Про нас» -

```

114 .content {
115   max-width: 1000px;
116   margin: auto;
117 }
118
119 .content p {
120   font-size: 18px;
121   color: ☐ #555;
122 }
123
124 .content img {
125   display: block;
126   max-width: 100%;
127   height: auto;
128   margin: 20px auto;
129   border-radius: 10px;
130 }

```

Рисунок 3.4 – Визначення стилів всіх блоків класу .content і дочірніх елементів

Як видно з рис. 3.4, для всіх блоків класу .content визначено однакову ширину і поля, так само всім дочірнім елементам з блоків тексту і зображень. Це свідчить про продуманий дизайн та роботу з раціоналізацією коду. За таким же принципом описані властивості блоків, які належать класу .grid-item (в розділі «Про нас» таких 3 згідно рис. 3.3). Зміст властивостей показано на рис. 3.5.

```

139 .grid-item {
140     background-color: #ffffff;
141     padding: 20px;
142     border: 1px solid #c8e6c9;
143     border-radius: 10px;
144     text-align: center;
145     transition: transform 0.3s;
146 }
147
148 .grid-item:hover {
149     transform: translateY(-10px);
150 }
151
152 .grid-item img {
153     width: 100%;
154     height: auto;
155     border-radius: 10px;
156     margin-bottom: 15px;
157 }

```

Рисунок 3.5 – Таблиці стилів для кожного з блоків у секції «Про нас» і дочірніх елементів у Visual Studio Code

Часто буває необхідно якимось чином трансформувати візуальне подання елемента (масштабувати, повернути, перемістити) і при цьому ніяк не торкнутися сусідніх елементів у документі. Для таких перетворень використовується властивість `transform`. Як значення виступають різні функції трансформації: `rotate`, `translate`, `scale`, `skew` та інші. Як видно з рис. 3.5, застосовано спеціальні функції трансформації блоків при наведенні мишею:

- властивість `transition: transform 0.3s;` - використовується для того, щоб плавно змінити CSS-властивості між двома станами елемента;
- функція `transform: translateY(-10px);` - використана для зміщення елемента вздовж вісі Y (по вертикалі).

В цілому, ту ж роботу виконують CSS-властивості `top`, `right`, `bottom`, `left` - наприклад, абсолютно (`position: absolute`) або відносно (`position: relative`) позиціонованих елементів. Але є низка важливих відмінностей: елемент позиціонується щодо відповідних сторін батька. Тобто `left: 20px` помістить елемент на 20 пікселів щодо лівої межі батька, а `translate(20px)` помістить

елемент праворуч щодо того місця, де той перебував до трансформації. В цілому, position та translate чудово поєднуються один з одним. Позиціонування краще використовувати для початкового розташування елемента на сторінці, а translate застосовувати, якщо потрібно додати анімацію руху. Приклад зміщення середнього блоку вгору при наведенні показано на рис. 3.6.

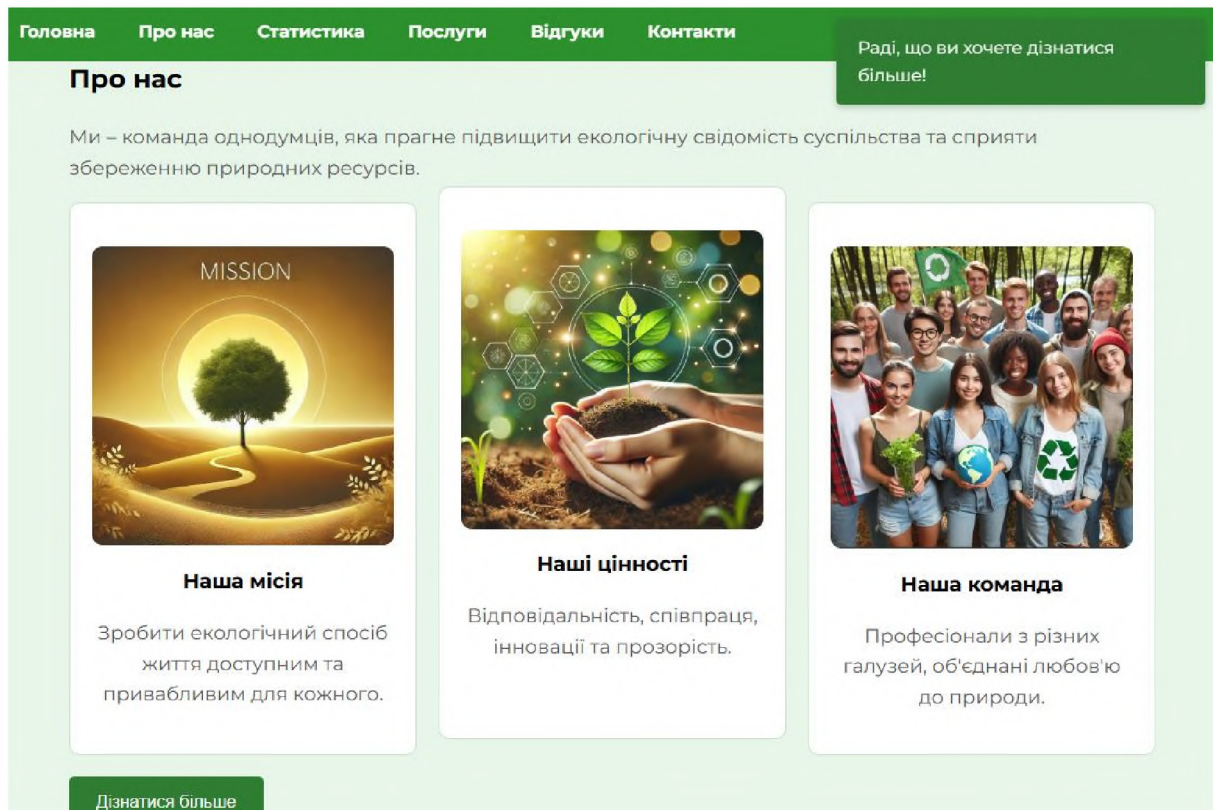


Рисунок 3.6 – Приклад анімації середнього блоку в секції та активація повідомлення

Всередині секції розміщена активна кнопка, яка викликає появу повідомлення і може супроводжуватися іншими діями (див. рис. 3.6). Подібні властивості має секція після статистики – «Наші послуги». HTML-код секції записаний в рядках 103-142 (див. додаток Б). Однак властивості використовують ті ж класи, які були описані вище (див. рис. 3.5): `class="content"`, `class="grid-item"`, дочірні елементи, які належать цим класам - `.grid-item img{}`, `.content p{}`. В більшості секцій є спеціальна кнопка, яка активує повідомлення. Вона знаходиться в окремому вкладеному блоці кожної секції, має однакове

оформлення (рис. 3.7) і активується подією onclick (рис. 3.8). Текст повідомлення може варіюватися.

```

91  /* Стили кнопок */
92  button {
93      padding: 15px 25px;
94      font-size: 16px;
95      margin: 20px 0;
96      cursor: pointer;
97      background-color: #2e7d32;
98      color: white;
99      border: none;
100     border-radius: 5px;
101     transition: background-color 0.3s;
102 }
103
104 button:hover {
105     background-color: #1b5e20;
106 }

```

Рисунок 3.7 – Опис спільних властивостей кнопок на вебсайті

```

138     </div>
139     <button onclick="showMessage('Деталі наших послуг надіслані на вашу пошту!'"
140     </div>
141 </section>

```

Рисунок 3.8 – Фрагмент HTML-коду з представленням розміщення кнопки

Загалом, стилізація сторінки є важливою частиною веброзробки, оскільки вона визначає зовнішній вигляд відображення вмісту вебсайту, досягнення дизайну інтерфейсу (UI-дизайн) та дизайну поведінки при використанні (UX-дизайн). Для цього в роботі використані такі CSS-стили, які дозволяють встановлювати різні параметри дизайну: такі як кольори, розміри, форми, типографіку та інші [14]. В розробленому проєкті використано зовнішній файл стилів, який підключається до сторінки за допомогою тегу <link>. Цей підхід дозволяє робити код більш читабельним та легшим для управління, завдяки викоистання універсальних класів для подібних секцій, блоків, а також властивості каскадності для дочірніх елементів. Все це робить код більш професійним і оптимальним. Повний лістинг коду стилів наведено в додатку В.

3.2 Технологічні аспекти організації оптимальної навігації інформаційного вебсайту та адаптивного дизайну

Зручна і продумана навігація відіграє чи не найголовнішу роль у зручному користуванні й затримці користувача на сторінці односторінкового вебсату.

До навігаційних елементів належать пункти горизонтального або вертикального меню, кнопки, стрілки, повзунок для скролінгу та деякі інші засоби. Розглянемо детальніше для прикладу облаштування горизонтального меню та технологій забезпечення його властивостей як елемента інтерфейсу. Горизонтальне меню містить 6 пунктів (див. рис. 3.5). Меню створено за допомогою семантичних тегів `<nav>` і розміщено одразу після `<body>`. Код меню показано на рис. 3.9.

```

11 <body>
12
13 <nav>
14   <a href="#home">Головна</a>
15   <a href="#about">Про нас</a>
16   <a href="#statistics">Статистика</a>
17   <a href="#services">Послуги</a>
18   <a href="#reviews">Відгуки</a>
19   <a href="#contact">Контакти</a>
20 </nav>
21
22 <section id="home">
23   <header>

```

Рисунок 3.9 – Структура HTML-коду меню сайту

Властивості пунктів меню описані відповідними ідентифікаторами, які розміщено в окремому файлі CSS-правил. Значення застосованих властивостей показано в коді на рис. 3.10.

```

84 /* Стили секцій */
85 #about {background-color: #e8f5e9;}
86 #statistics {background-color: #d0f0c0;}
87 #services {background-color: #c8e6c9;}
88 #reviews {background-color: #b2dfdb;}
89 #contact {background-color: #a5d6a7;}

```

Рисунок 3.10 – Фрагмент CSS-коду для опису стилів секцій

Перехід по кожному пункту меню переводить користувача на відповідний блок сторінки – секцію, якій присвоєно ідентифікатор. Зовні кожна секція вирізняється кольором, вірніше відтінком. Однак, до кожного пункту меню прив’язана не лише секція, а й метод переходу, інтерактивна подія на вебсайті.

Набір стилів для опису поведінки навігаційної панелі показано на рис. 3.11.

```

14  /* Стили навігації */
15  nav {
16      position: fixed;
17      top: 0;
18      width: 100%;
19      background-color: rgba(34, 139, 34, 0.95);
20      overflow: hidden;
21      z-index: 1000;
22      height: 60px;
23  }
24
25  nav a {
26      float: left;
27      display: block;
28      color: #fff;
29      text-align: center;
30      padding: 20px 20px;
31      text-decoration: none;
32      font-weight: bold;
33      transition: background-color 0.3s;
34  }
35
36  nav a:hover {
37      background-color: #3275a8;
38  }
39
40  /* Відступ для контенту, щоб не перекривався навігацією */
41  section {
42      padding: 80px 20px;
43      min-height: 100vh;
44      padding-top: 100px; /* 60px навігація + 40px відступ */
45  }

```

Рисунок 3.11 – Опис стилів панелі навігації на сайті

Аналізуючи зміст властивостей кожного із записаних стилів, можна зробити висновок, що перший теговий стиль `nav` стосується всієї панелі і мстить властивість закріплення на екрані, ширину і висоту панелі, фон. Наступні стилі використовують властивість каскадності і описують вигляд і поведінку кожного запису в меню `nav a` : розміщення зліва направо, текст вирівнювати по центру;

колір фону і колір тексту (білий). Остання частина описує зміну кольору конкретного пункту при наведенні мишкою, використовуючи функцію `hover` в записі `nav a: hover`.

За м'якість прокручування всієї сторінки відповідає властивість (рис. 3.12):

```

9      /* Плавне прокручування */
10     html {
11         |         scroll-behavior: smooth;
12     }

```

Рисунок 3.12 – Запис властивості плавного прокручування всієї вебсторінки

Більше динаміки при взаємодії користувача, звичайно, додає JavaScript. Наприклад, плавне прокручування з урахуванням фіксованої навігації показано на рис. 3.13.

```

JS script.js > showMessage
1      // Плавне прокручування з урахуванням фіксованої навігації
2      document.querySelectorAll('nav a').forEach(anchor => {
3          |         anchor.addEventListener('click', function(e) {
4              |             e.preventDefault();
5              |             const targetId = this.getAttribute('href').substring(1);
6              |             const targetSection = document.getElementById(targetId);
7              |
8              |             const navHeight = document.querySelector('nav').offsetHeight;
9              |
10             |             window.scrollTo({
11                 |                 top: targetSection.offsetTop - navHeight,
12                 |                 behavior: 'smooth'
13             |             });
14         |         });
15     });

```

Рисунок 3.13 – Набір функцій підтримки навігації в JS

Додатковий ефект, який досягнуто при завершенні розробки вебсайту , було запровадження спливаючого повідомлення, яке з'являється в правій частині на певний період часу. Повний код представлено на рис. 3.14.

```

17 // Функція для відображення спливаючого повідомлення
18 function showMessage(message) {
19     const container = document.getElementById('notification-container');
20
21     // Створюємо новий елемент повідомлення
22     const notification = document.createElement('div');
23     notification.classList.add('notification');
24     notification.textContent = message;
25
26     // Додаємо повідомлення до контейнера
27     container.appendChild(notification);
28
29     // Видаляємо повідомлення після закінчення анімації
30     setTimeout(() => {
31         notification.remove();
32     }, 5000); // 5 секунд
33 }

```

Рисунок 3.14 – Три частини обробки повідомлення за допомогою функцій JavaScript (зміст – в коментарях)

Отже, основними технологічними особливостями при формуванні оптимального навігаційного блоку вебсайту є використання опису елементів навігації в спеціальних файлах стилі та поведінки, тобто завдяки застосуванню CSS і програмування на JavaScript. Подальше удосконалення цієї частини роботи може, безумовно, включати, згортання меню в гамбургер, дослідження перебудови блоків при перехід до адаптивності.

Адаптивний дизайн – це підхід до розробки вебсайтів та програм, що дозволяє створювати інтерфейси, які можуть адаптуватися до різних пристроїв та екранів. Це означає, що вебсайт буде відображатися оптимально на всіх пристроях, включаючи комп'ютери, ноутбуки, планшети, смартфони і т.д.

Процес розробки адаптивності починається з планування та проєктування інтерфейсу користувача, який буде легко читаним і використовуватися на всіх пристроях. Потім розробники використовують медіазапити CSS для створення різних версій вебсайту або програми для кожного пристрою. Медіазапити дозволяють визначати різні характеристики пристрою, такі як ширина екрану та орієнтація, і відповідно змінювати зовнішній вигляд та розташування елементів на сторінці.

Адаптивний дизайн необхідний для забезпечення максимальної зручності використання вебсайту або програми для всіх користувачів, незалежно від того, який пристрій вони використовують. Це також допомагає покращити SEO-оптимізацію, оскільки Google рекомендує використовувати адаптивний дизайн для покращення взаємодії з користувачами та підвищення якості сайтів [42]. Опис адаптивності в кінці файлу стилів представлено на рис. 3.15.

```

281  /* Адаптивність */
282  @media (max-width: 768px) {
283      header h1 {
284          |     font-size: 48px;
285      }
286      header p {
287          |     font-size: 20px;
288      }
289      .stat-item h3 {
290          |     font-size: 36px;
291      }
292  }
293
294  @media (max-width: 600px) {
295      nav a {
296          |     float: none;
297          |     width: 100%;
298          |     text-align: left;
299          |     padding-left: 20px;
300      }
301      nav {
302          |     height: auto;
303      }
304      .grid-container {
305          |     grid-template-columns: 1fr;
306      }
307      .reviews-container {
308          |     flex-direction: column;
309      }
310  }

```

Рисунок 3.15 – Властивості адаптивності для вебсату

Як видно (див. рис. 3.5), для адаптивності були використані @media, Flex властивості, різні комбінації значень max-widht. Загалом, щоб вирішити проблему адаптивності вебсторінки (див. рис. 3.15, а також додаток В), виконувались наступні технології та прийоми, властивості:

- відносні одиниці для вимірювання розмірів елементів сторінки (наприклад, відсотки, em, rem), а не фіксовані значення. Це дозволило елементам сторінки пропорційно змінюватися в залежності від розміру екрана.

- медіа-запити: дозволяють встановлювати різні стилі для елементів сторінки в залежності від розміру екрана (рис. 3.5).

- різні зображення для різних розмірів екранів - це дозволить зменшити завантаження сторінки та зберегти трафік.

На завершення було проведено тестування вебсторінки на різних пристроях, щоб переконатися, що вона коректно відображається на всіх пристроях та не має проблем з адаптивністю.

Підключення бібліотек JavaScript через API (Application Programming Interface), який є інтерфейсом між різними програмними продуктами, що дозволяє їм обмінюватися даними, дозволяє підключити зовнішні бібліотеки JS безпосередньо з сервера. Це зменшує час завантаження сторінки та забезпечує кращу продуктивність вебсайту. При підключенні бібліотек JavaScript через API, сервер передає запит на зовнішній API та отримує відповідь, яка містить код бібліотеки. Цей код можна додати до сторінки вебсайту і використовувати функціональність бібліотеки безпосередньо на сторінці вебсайту.

3.3 Економічне оцінювання вартості проєкту

Проведемо економічну оцінку і розрахунки щодо вартості виконання робіт з проєктування та розроблення такого типу вебсайту, а також хостингу.

Оцінка вартості розробки сайту у спеціалізованих компаніях залежить від кількості робочих годин і тарифу, встановленого за 1 годину роботи. Тариф залежить від складності сайту і кваліфікації розробника. Наприклад, створення інтерактивного сайту-лендінгу із 6 секцій із використанням динамічних ефектів може займати приблизно 60 годин роботи. Вартість години роботи фрілансерів або вебстудій в Україні оцінюється в діапазоні 10-20 доларів США [48].

Економічне обґрунтування вартості вебсайту здійснюється з урахуванням програмних засобів, розробки сайту, оренди доменного імені та його хостингу. Усі детальні розрахунки проводилися в декілька етапів.

1. Оцінка затрат на інформаційні технології: визначається обсяг інвестицій в інформаційні технології, необхідний для досягнення поставлених цілей. Оцінка витрат на ІТ має кілька складових. Оцінка витрат за проектом передбачає визначення усіх капітальних і поточних витрат, пов'язаних з впровадженням і використанням інформаційних технологій, тобто:

а) оцінка прямих витрат на впровадження проекту. Визначення прямих витрат розраховується за формулою:

$$V_{\Pi} = V_{T3} + V_X + V_{BC3} + V_{BC\Pi} + V_y + V_{\Pi\Pi} + V_I, \quad (3.1)$$

де V_{T3} – витрати на технічне забезпечення, грн;

V_X – витрати на хостинг та придбання домену, грн;

V_{BC3} – витрати на соціальні заходи, грн;

$V_{BC\Pi}$ – витрати, за послуги сторонніх організацій, наприклад, придбання унікального контенту, який буде розміщено на сторінках сайту, грн;

V_y – витрати на управління інформаційними технологіями, тобто витрати на послуги провайдера на 12 місяців, грн;

$V_{\Pi\Pi}$ – витрати на програмне забезпечення (наприклад, тарифний план Figma), грн;

V_I – реклама для сайту, грн.

б) Оцінка непрямих витрат на впровадження проекту (V_H) із урахуванням ризиків: дозволяє розрахувати витрати від простоїв, витрат на усунення таких збоїв та інші витрати і розраховуються за формулою:

$$V_H = V_{H1} + V_{H2} \quad (3.2)$$

де V_{H1} – витрати, пов'язані з простоями через зупинку роботи інформаційної системи, у даному випадку, простій доменного імені, грн;

V_{H2} – витрати, пов’язані з технічним фактором (поломка ноутбуку, відсутність мережі інтернет тощо), грн.

в) оцінка витрат на утримання інформаційних технологій за увесь період життєвого циклу проєкту. Етап допомагає передбачити щорічну величину витрат на утримання інформаційних технологій на протязі їх використання і розраховуються за формулою:

$$V_{УТР} = V_{ПС} + V_{ВСЗ} + V_{ПСП} + V_I, \quad (3.3)$$

де $V_{ПС}$ – витрати на підтримку та удосконаленні сайту 3 рази на рік, грн;

$V_{ВСЗ}$ – витрати на відрахування на соціальні заходи (не входять в реалізацію сайту), грн;

$V_{ПСП}$ – витрати на послуги, які надають інші підприємства, наприклад, оновлення контенту на сайті, та реклама будуть здійснюватися раз на рік, грн;

V_I – інші витрати, зокрема, витрати на інтернет, грн.

Сумарна величина витрат за проєктом розраховується за формулою:

$$V_{IT} = V_{П} + V_{H} + V_{УТР} \quad (3.4)$$

2. Оцінка вигоди від впровадження інформаційних технологій. На даному етапі проводиться оцінка очікуваної вигоди від впровадження. Існує кілька видів ефектів від впровадження інформаційних технологій.

Для інформаційного сайту важливим є кількість унікальних користувачів, перегляду його вмісту та нових клієнтів. Для проведення маркетингового аналізу та підбору ключових слів також необхідно зібрати семантичне ядро [51]. Найчастіше для цього застосовується сервіс Google Ads. Ключові слова можуть бути низькочастотними, слова, які доволі рідко запитуються в пошукових системах, середньочастотні, такі запити шукають до 5 тисяч разів на місяць та високочастотні – словосполучення, які найчастіше шукають користувачі (до 10 тисяч разів на місяць). Грамотна побудова семантичного ядра сайту є головним моментом в оптимізації та його ефективності.

Оцінка вартості витрат на розроблення вебсайту наведена в табл. 3.1 для односторінкового вебсайту, в складі якого 6 секцій, із подальшим хостингом та аудитом на основі актуальних даних про середню вартість робіт та послуг.

Таблиця 3.1 – Розрахунок прямих витрат на розроблення вебсайту

№ з/п	Види витрат на розробку і впровадження технології	Витрати часу, одиниці	Грошовий еквівалент робіт, грн/од. часу	Сумарна вартість, грн
1	Розроблення макету вебсайту в програмі Figma (ПЗ безкоштовно)	16 год.	300	4800
2	Розроблення сайту для хостингу користування	60 год.	410	24600
3	Технічний аудит вебсайту перед хостингом та оптимізація	16 год.	300	4800
4	Оплата домену (хостинг) від компанії TheHost, пакет «Мій сайт» на рік (500 Мб) [50]	1 рік	749	749
5	Вартість домену (спеціальний міжнародний загальний домен in.net) на рік	1 рік	199	199
6	Вартість послуг інтернет за період роботи	1 рік	3600	3600
7	Сумарна вартість прямих витрат індивідуального замовлення вебсайту	$=(п.1+п.2+ п.3+ п.4+ п.5)$		35148
8	Вартість вебсайту разом із утриманням за рік	$=(п.6 +п.7)$		38648

Отже, оцінювання вартості розробки методом витрат, при якому враховані прямі витрати (див. табл. 3.1, п.1-5) без витрат на утримання (п. 6), згідно розрахункових даних становить близько 35,148 тис. грн із врахованим хостингом за рік. Якщо організація оплачує інтернет, то сума витрат складає 38648 грн. При цьому чиста вартість роботи 34200 грн.

Розраховані дані носять оціночний наближений характер, виконані за прийнятими методиками, але можуть бути уточнені в конкретних випадках.

ВИСНОВКИ

Підсумування виконаних в даній роботі досліджень тенденцій розвитку вебтехнологій, проектування та розроблення вебсайту для обраної предметної області, а також попередній аудит його основних характеристик, розрахунок майбутньої ефективності дозволяють зробити наступні висновки.

1. Представлення інформації про діяльність підприємств певної галузі в мережі інтернет є дуже важливим, оскільки має позитивний вплив на просування бренду, залучення нових прихильників, розширення цільової аудиторії і підвищення репутації. Інформаційні технології значно змінили організацію діяльності, зокрема, громадських і просвітницьких організацій, оскільки допомагають забезпечити більший обсяг цільової аудиторії.

2. Аналіз представлення інформації на різних вебсайтах в мережі інтернет на екологічну тематику дозволяє резюмувати, що в основному вебсайти супроводжують інформаційну діяльність окремих організацій або установ, які займаються питаннями популяризації теми по збереженню довкілля. При цьому якість таких сторінок не завжди відповідає місії, не мають зворотного зв'язку, не регулярно оновлюють інформацію. Відтак, якісні вебсайти слугуватимуть привертанням більшої уваги до даної проблематики, сприятимуть реалізації колективних проєктів.

3. Розвинуті вебтехнології дозволяють розробляти вебсайти високої якості із використанням базових технологій HTML, CSS, мов програмування для веброзробок, а також спеціалізованих середовищ графічного проєктування та середовищ розробки й редагування кодів, Visual Studio Code, Figma. Розроблено інтерактивний вебсайт з динамічними елементами (трансформація зображень тощо), який має зручну навігацію та раціональний професійний код.

4. Показано прийоми технічного аудиту вебсайту та SEO-аудиту за допомогою спеціальних програм Google Search Console та PageSpeed Insights на етапах підготовки та після пробного хостингу. Отримано достатньо високий рівень оцінювання оптимізації сайту.

5. До основних показників, що впливають на розрахунок економічної ефективності створення сайту відносять:

- стартові витрати на створення сайту;
- експлуатаційні витрати, пов'язані із забезпеченням якісного функціонування сайту;
- співвідношення цих витрат;
- показники для оцінки ефективності інвестиційних проєктів.

Значення витрат на розроблення подібного вебсайту дорівнює 35,148 тис. грн без урахування витрат на утримання.

Реалізація представленого проєкту сайту може бути запланована як макет, придатний до модернізації і вдосконалення, для громадських організацій, інших спільнот, які зацікавлені у висвітленні проблем охорони довкілля.

Для розробника такий проєкт може стати частиною власного портфоліо та об'єктом самовдосконалення і розвитку власної майстерності в майбутньому.