

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

Освітньо-професійна програма Інформаційні управляючі системи  
Спеціальність 126 Інформаційні системи та технології  
Ступінь вищої освіти Бакалавр

## **КВАЛІФІКАЦІЙНА РОБОТА**

на здобуття ступеня вищої освіти бакалавр

на тему:

**«Проектування інтерактивного вебдодатку для супроводу вивчення  
іноземних мов»**

Виконав: здобувач вищої освіти  
за освітньою програмою  
Інформаційні управляючі системи  
спеціальності 126 Інформаційні  
системи та технології  
ступеня вищої освіти бакалавр  
групи 126ІСТ\_бд\_2021  
Липоватий Владислав Віталійович  
Керівник: Слюсар Вадим Іванович  
Рецензент: Муравльов Володимир  
В'ячеславович

**Полтава – 2025 року**

## ВСТУП

У сучасному світі глобалізації та цифрових технологій знання іноземних мов стало необхідністю. Водночас традиційні методи навчання не завжди відповідають потребам сучасного користувача, який очікує швидкого доступу до знань, індивідуального підходу та інтерактивності.

*Актуальність* розробки вебдодатків для інтерактивного навчання іноземним мовам розглядається в кількох аспектах. Вебдодатки забезпечують гнучкість навчального процесу. Користувачі можуть навчатися у зручний для себе час і в будь-якому місці, маючи доступ до платформи з комп'ютера чи мобільного пристрою. Це особливо важливо для людей з напруженим графіком або тих, хто проживає у віддалених населених пунктах.

Інтерактивні можливості вебдодатків стимулюють мотивацію та залучення користувача. Гейміфікація, інтерактивні вправи, аудіо- та відеоматеріали, віртуальні діалоги й тести сприяють кращому засвоєнню матеріалу порівняно з пасивним читанням або прослуховуванням. Такі інструменти дозволяють залучити різні канали сприйняття інформації (відео, аудіо, текстовий контент, тестування) та зробити процес навчання більш захопливим і ефективним. Нарешті, сучасні технології дозволяють адаптувати навчальний контент до рівня знань та потреб конкретного користувача. Завдяки вбудованим алгоритмам, штучному інтелекту та аналітиці прогресу вебдодатки можуть пропонувати персоналізовані траєкторії навчання, виявляти слабкі місця та рекомендувати відповідні матеріали для їх подолання.

Навчальні вебдодатки мають бути зручними для навігації, доступними для використання, мати лаконічний привабливий дизайн, видимими для пошукових систем. Бажано бути адаптованими для завантаження на мобільні пристрої без зайвих зусиль. Багато посібників є досить інформативними, часто складаються з докладних інструкцій і прикладів коду, які є корисними для швидкого вирішення проблеми або відповіді на запити.

*Метою* кваліфікаційної роботи є створення сучасного інтерактивного адаптивного вебдодатку для супроводу вивчення однієї з іноземних мов, розробленого на основі класичних технологій HTML&CSS, JavaScript.

*Об'єктом* дослідження є процес розроблення вебдодатку із забезпеченням функціональності та відповідного дизайну на основі глибокого розуміння властивостей обраних технологій.

*Предметом* дослідження є практичні аспекти використання вебтехнологій HTML та CSS, мови програмування JavaScript, оригінального комбінування при розробці вебдодатків із заданими властивостями.

*Методами* дослідження є інформаційно-пошуковий, аналітико-синтетичний, порівняльних характеристик, комбінаторний, моделювання професійних ситуацій, оцінювання економічної ефективності та інші.

*Завданнями* кваліфікаційної роботи є:

- дослідити сучасні тенденції дизайну та засобів досягнення функціональності й властивостей навчальних вебдодатків;
- провести порівняльний аналіз наявних типів вебсайтів у предметній області;
- описати етапи підготовки й розробки вебсайтів заданої тематики;
- проаналізувати та застосувати на прикладах інструментальні засоби та методи розроблення інтерактивного вебдодатку для навчання однієї з європейських мов;
- провести технічний аудит оптимальності досягнутого функціоналу та оцінити економічні показники веброботки.

Практична значущість полягає у розробці сучасного вебдодатку, який висвітлює весь спектр навчальних послуг, які направлені на взаємодію користувача з необхідним контентом через інтерактивний вебдодаток. Дизайн вебінтерфейсу виглядає як завершений продукт, який в той же час може бути масштабованим на різні рівні курсів, а також переобленим на інші іноземні мови. Додаток є доволі універсальний, має адаптивні властивості, і його можна використати для просування будь-яких інших послуг.

Результати роботи апробовані в рамках XXI науково-практичної студентської конференції 16 квітня 2025 р. та опубліковані у вигляді тез доповіді, посилання на які включено до списку літератури.

Структура кваліфікаційної роботи логічно пов'язана з задачами досліджень, проектування та розробки, містить перелік умовних познач, вступ, три розділи основної частини, висновки, список використаних джерел. Загальний обсяг текстової частини кваліфікаційної роботи складає 50 сторінок формату А4. Вона містить 21 рисунок і 4 таблиці. В роботі використано 42 бібліографічних посилання.

## РОЗДІЛ 1

### ТЕОРЕТИЧНИЙ АНАЛІЗ СУЧАСНОГО СТАНУ ТЕХНОЛОГІЙ РОЗРОБКИ ВЕБДОДАТКІВ НАВЧАЛЬНОГО СПРЯМУВАННЯ

#### 1.1 Етапи удосконалення мови розмітки гіпертексту HTML як відправної технології розробки вебдодатків в історичному аспекті

Більшість користувачів вже неодноразово чули такі літери як HTML – вони сходять з вуст будь-якого користувача-початківця, який тільки-но почав свій шлях у вебтехнологіях. Це скорочена аббревіатура від Hypertext Markup Language, що в перекладі з англійської означає мова розмітки гіпертексту. Більшість сучасних інтернет-технологій заснована на мові HTML, що давно використовується і обговорюється. Вона призначена для розмітки та форматування документів, розміщених на вебсторінках. На основі досвіду спільноти вебпрограмістів добре підтверджена думка, що HTML є основною рушійною силою створення вебсторінок [1].

Мова почала набувати своїх перших функцій іще у 1986 році [1]. Поштовхом стало ухвалення Міжнародною організацією зі стандартизації (ISO) стандарту ISO-8879 - Standard Generalized Markup Language, або скорочено - SGML. Він постачається із приміткою, що SGML призначений для структурної розмітки тексту. Варто зазначити, що опис зовнішнього вигляду документа не передбачався [2]. Ця мова не надавала жодного набору структурних елементів, необхідних для конкретних ситуацій чи розмітки тексту. Натомість, SGML описував виключно синтаксис, що використовується для створення основних елементів розмітки. Через деякий час вони отримали, добре відому сьогодні назву – теги. Звідси зрозуміло, що SGML не призначався для розмітки тексту.

У 1991 р. Європейський інститут фізики елементарних частинок заявив про необхідність створення засобів передачі гіпертекстових даних у режимі онлайн. Незважаючи на те, що він не сильно оновлювався, мова SGML не була

забута і стала основою, на якій була побудована добре відома мова гіпертекстової розмітки HTML.

HTML – це мова, яка описує структуру сторінок документа. Вона дозволяє формувати звичайний текст у абзаци, заголовки, списки тощо, а також створювати посилання на інші сторінки. Це текстова мова, у якій інструкції форматування (так звані теги) вбудовані в розділи документів, які містять певну інформацію. Теги повідомляють браузеру, як формувати та відображати інформацію на екрані.

У 1989 р. Тім Бернс-Лі, тоді маловідомий британський вчений, який працював у Європейському центрі ядерних досліджень (CERN), реалізував ідею використання гіпертексту у вигляді World Wide Web заснував HTML. Його мета полягала в тому, щоб спростити обмін документами між вченими в різних університетах [3]. Однак його широкий успіх швидко перетворив його на основу інтернету, який ми знаємо сьогодні. Мова гіпертекстової розмітки HTML стала офіційною мовою Всесвітньої павутини для створення та публікації вебсторінок. Його основною функцією є включення заголовків, тексту, таблиць, фотографій, відеокліпів, звуків та інших зовнішніх об'єктів у вебдокументи. По суті, HTML є основою сучасної веброзробки, стійкою платформою, на якій нашаровуються безліч інших технологій.

Вибір спеціалізованих HTML-редакторів або інтегрованих середовищ розробки та конвертерів – це перший крок до створення HTML-документів. Думки в тому, яка програма краще, сильно різняться; зрештою це залежить від особистих уподобань та ідей автора для побудови ефективної роботи. Проте, теги є центральною концепцією, якої не можна уникнути. Вони надають мітки, які визначають, як браузер представляє різні види інформації, такі як шрифти, абзаци, заголовки, зображення та багато іншого. Враховуючи важливість пошукового просування для сайтів, освоєння тегів особливо важливе.

По своїй суті HTML – це набір тегів що слідує певному порядку, починаючи з кутових дужок «<» і закінчуючи знаком «>» [4]. Початковий та кінцевий теги, зазвичай, становлять певну структуру, причому останній має

слеш перед словом, наприклад «<p>» та «</p>». Фактична інформація перебуває між прописаними тегами. Вказавши, як слід декодувати документ, браузер автоматично визначає це при отриманні. Таким чином, дуже важливо, щоб початковий тег документа передав його кодування. Цей вихідний тег є доказом того, що він використовувався під час створення текстового або графічного сегмента.

Для документів, що містили близько сорока тегів HTML версії 1.2, не було описано жодної фізичної думки. Хоча SGML був його предком, логічна та структурна текстова розмітка залишалася у центрі уваги HTML. Проте, добірка тегів дала натяк на те, як сторінка буде сприйматися у фізичному сенсі.

У своїй пропозиції 1989 р. для інтернету Тім Бернерс-Лі включив діаграму, що зображує приклад семантичної мережі, заснованої на іменованих ресурсах з позначеними посиланнями між ними (рис. 1.1) [5].

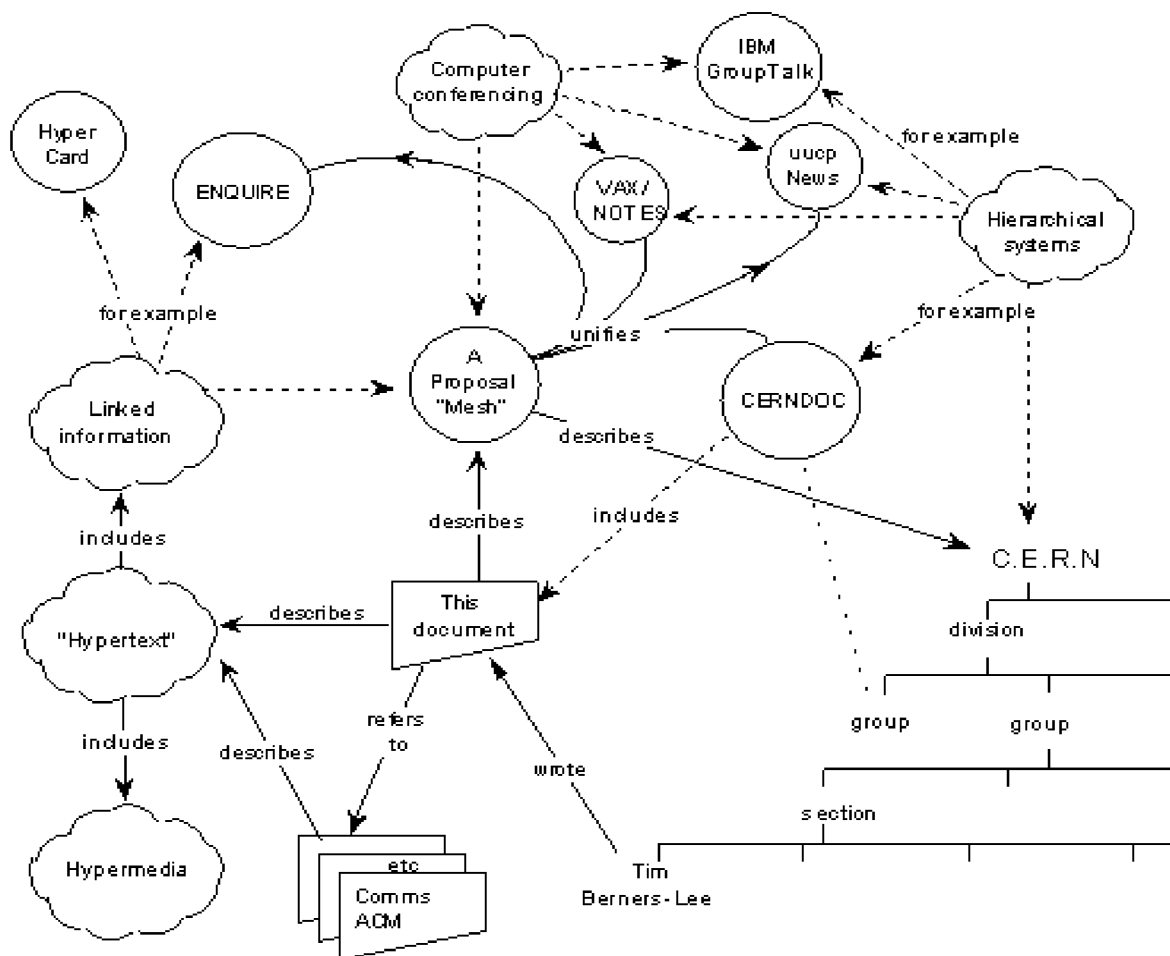


Рисунок 1.1 – Структура опису ресурсів семантичної мережі та пов'язані дані

Ця ідея була розвинена W3C (World Wide Web Consortium – консорціум всесвітньої павутини) в рамках W3C Resource Description Framework (RDF), де URL-адреси використовуються як для ресурсів, так і для міток посилань. Кожне посилання (також відоме як трійка), таким чином, складається з URL-адрес для суб'єкта, предиката та об'єкта відповідно. URL-адреси діють як ім'я, так і як засіб отримання додаткової інформації шляхом розкладання URL-адреси за допомогою HTTP GET-запиту на URL-адресу. Протягом багатьох років W3C розробив набір стандартів навколо RDF.

Деякі стандарти стосуються використання RDF для визначення моделей, наприклад, RDF Core, RDF Schema та OWL. Інші визначають формати обміну даними для RDF, наприклад, RDF/XML, N-Triples, Turtle, TriG та JSON-LD. Платформа зв'язаних даних (LDP) визначає, як використовувати HTTP для читання та запису трійок. SPARQL – це мова запитів та оновлення для RDF, аналогічна SQL для реляційних даних. SHACL надає засоби для вираження обмежень дійсності набору трійок.

1995 р. W3C створив HTML версії 2.0 після року інтенсивної роботи [6]. Водночас, обговорювалися можливості версії 3.0. Хоча друга версія не дуже відрізнялася від першої, третя стала безперечним проривом. Вона включала такі основні нововведення:

- розмітку математичних формул;
- теги для створення сторінок;
- вставку малюнків, що обтікаються текстом;
- приміток і т. д.

Компанія Netscape Communications, компанія що запустила перший комерційний браузер – Netscape Navigator, запропонувала значне розширення HTML-тегів [7]. Ця ініціатива була спрямована на підвищення візуальної привабливості документів, але також суперечила початковим принципам мови. Незабаром було створено HTML версію 3.2. Вона була спрямована вже на новостворений Microsoft Internet Explorer [8]. До 1999 р. ця версія HTML була єдиним мовним стандартом розробки інтернет-проектів. Тим не менш,

розробки в напрямку йдуть дуже активно. За допомогою HTML можлива деяка впорядкованість елементів розмітки для всіх браузерів, але можливості і цієї версії згодом стає недостатньо.

У 1998 р. була прийнята нова версія HTML – 4.0, а в 1999 р. її наступник – версія 4.01. Вона пропонує більш кращу крос-браузерну та крос-платформну продуктивність. Але ж тільки на той час і для більш адаптивної розробки її не вистачало.

Актуальність візуального дизайну гіпертекстових сторінок зростала, і просто покладатися на HTML було недостатньо [9]. Тому W3C розробила незалежну систему, яка відповідала HTML та дозволяла відобразити візуальний дизайн документів [21]. Результатом стало створення каскадних таблиць стилів (CSS), ієрархічних специфікацій стилів з унікальним синтаксисом, структурою та можливостями [2].

До того, W3C розпочав розробку мови HTML5, набір функцій для створення структури вебсайту все ще був досить невеликим і обмеженим. Наприклад для додавання мультимедії, потрібно було використовувати пропрієтарні компоненти (Flash і т.п.).

Набір тегів для маркування був невеликий. Структурні результати нечитані і складні. Завдання підтримки та розвитку ресурсів стало неймовірно складним. Вирішення цих проблем та розширення функціоналу вебсторінок – основне завдання нової технології HTML5, яка на даний час підтримується всіма сучасними браузерами [8].

Оскільки потреба у візуально привабливих гіпертекстових сторінках зросла, HTML5, новий стандарт розмітки для сучасного вебконтенту був розроблений і, нарешті, випущений у 2014 році, став доступний для всіх розробників. Поступово багато сучасних вебсайтів прийняли цей інноваційний підхід через різні фактори.

Світ веброзробки зазнав радикальних змін з появою HTML5, оскільки він усунув усі обмеження попередньої версії. Крім того, тепер є безліч нових тегів, що дозволяють більш точно описувати елементи сторінки. Відмова від

Flash і подібних технологій стала можлива завдяки вбудованій в HTML5 можливості вбудовування мультимедійних об'єктів. Разом з цим зникли супутні проблеми, такі як низька продуктивність та вразливість коду.

Сьогодні HTML5 дозволяє створювати складні вебсайти, що включають мультимедійні та інтерактивні функції, не покладаючись на інші рішення. Мова повністю оснащена вбудованими функціями, такими як відео плеєри, засоби перевірки форм, графічні дизайнери та багато іншого, які поєднані з аспектами CSS, XHTML, DHTML та Ajax.

Багато старих тегів, у яких більше немає необхідності, було знято з виробництва, щоб робота HTML5 не стала надто складною. Але водночас набуття фундаментальних технологічних навичок стало набагато простіше. З легко засвоюваними та зрозумілими етикетками було введено свіжі теги.

## **1.2 Характеристика технологій, сумісних з HTML, та альтернативні методи веброзробки**

CSS (каскадні таблиці стилів) – це мова для опису зовнішнього вигляду вебдокументів, створених за допомогою мови розмітки HTML [10].

CSS – це широкомасштабна технологія, схвалена W3C та названа «вебстандартом». У 1990 р. необхідно було стандартизувати їх у деякі конкретні єдині правила завдяки вебдизайнерам та програмістам, які розробляють вебсайти.

Хокон Віум Лі, норвезький спеціаліст з інформаційних технологій, запропонував ідею каскадних таблиць стилів, працюючи у консорціумі W3C. На початку 1990-х у різних браузерів були свої стилі відображення вебсторінок. HTML швидко розвивався і зміг задовольнити всі потреби інформаційного дизайну, що тоді існували, тому CSS на той час не набув широкого поширення.

CSS3 – це остання версія популярної мови, яка використовується для оформлення вебсайтів, але до неї, звісно, було кілька попередніх версій. Перша версія CSS стала рекомендацією W3C 17 грудня 1996 року. Майже через два роки, 18 травня 1998 року, було прийнято CSS2. Оновлену версію CSS2, яка отримала назву CSS2.1, схвалили 8 вересня 2009 р. Версія CSS4 офіційно не вийшла, хоча знаходиться в розробці з 29 вересня 2011 р.

JavaScript – це мова програмування високого рівня, яка використовується для створення динамічних вебсторінок та програм. JavaScript дозволяє додавати інтерактивність на вебсторінки, такі як анімації, зміна вмісту сторінки без перезавантаження, обробка подій, валідація форм і багато іншого. Інтерактивні вебсторінки створюються за допомогою JavaScript у співпраці з HTML та CSS.

Брендан Ейх створив JavaScript у 1995 році, і з того часу він став популярною мовою програмування у всьому світі [11]. Завдяки Node.js цю мову можна використовувати для створення клієнтських або серверних програм. JavaScript, що знаходиться в браузерах або на серверах без компіляції, є інтерпретованою мовою. На сьогодні розробка JavaScript стала більш доступною, ніж будь-коли, завдяки безлічі бібліотек і фреймворків, які дозволяють створювати програми, використовуючи готові коди і алгоритми. Аналітичні компанії, наприклад StackOverflow публікують регулярні звіти [12], що підтверджують лідируючі позиції JS серед мов програмування, як серед професіоналів (додаток А, рис. А1), так і серед всіх, хто займається вебтехнологіями (додаток А, рис. А.2).

Гнучка і потужна мова, JavaScript здатний створювати різні вебзастосунки з його широким набором функцій. Підтримка функціонального програмування, об'єктно-орієнтованого програмування (ООП) та асинхронного програмування робить його надзвичайно цінним.

За допомогою цієї універсальної мови програмування застосовується творчий підхід до художніх творів, таких як мобільні програми, настільні програми, ігри та інше програмне забезпечення (ПЗ).

Сторонні бібліотеки та фреймворки, такі як Angular, Vue.js, Node.js, Express.js та React, надають безліч функцій для JavaScript [13]. Ці функції варіюються від маніпулювання масивами, математики, маніпулювання датами та рядками до інших функцій. Завдяки цим можливостям ми можемо легко створювати складні програми. Крім того, ми використовували різні стандартні бібліотеки, доступні для JavaScript.

Щоб з легкістю виконувати завдання, що повторюються, універсальний характер JavaScript дозволяє розробникам створювати ботів і сценарії для автоматизації таких завдань, як збір даних, моніторинг вебсторінок і обробка інформації.

Окрім використання засобів програмування, вебсайти можна створювати без будь-яких спеціальних знань у галузі кодування, використовуючи кілька конструкторів вебсайтів. Властивості найбільш поширених із них наведено в табл. 1.1.

Таблиця 1.1 – Характеристики поширених конструкторів вебсайтів

Назва конструктору	Зміст функціональних характеристик та властивостей
Wix	Один із найпопулярніших конструкторів сайтів. Він пропонує широкий вибір шаблонів, елементів дизайну, що настроюються, інтуїтивно зрозумілий інтерфейс, потужний редактор, інструменти SEO-оптимізації, вбудований хостинг і можливість створювати онлайн-магазини та блоги
Squarespace	Платформа для створення вебсайтів, яка включає понад 100 адаптивних шаблонів, а також функціональність по роботі з мультимедіа, інструменти аналітики, SEO-оптимізація та багато інших функцій.
WordPress	Одним із найпоширеніших конструкторів сайтів, який дозволяє створювати як прості, так і складні вебсайти. Він надає величезний вибір тем і плагінів, блог-функціональність, інструменти SEO, підтримку безлічі мов та багато інших можливостей.
Shopify	Одним із найкращих конструкторів онлайн-магазинів. Він надає готові шаблони для створення електронної комерції, інструменти оплати та доставки, безліч інструментів маркетингу, інтеграції соціальних мереж, звітність та багато іншого.
Webflow	Більш сучасним конструктором сайтів, який пропонує можливість створювати не тільки шаблонні сайти, але й унікальні вебдодатки. Він також надає інструменти для Проектування, редагування та управління контентом, інтеграцію з іншими додатками та сервісами та багато іншого.

Не вимагаючи навичок програмування, конструктори вебсайтів знаходяться практично у вільному доступі, кожен з яких має свої можливості, які можуть бути корисними для розробників певних типів та сайтів. Їх доступність в тому, що вони пропонують створення сайтів без знання мов вебпрограмування. Мінусом є платна основа таких конструкторів, що потрібна для повної реалізації та отримання доступу до повного інструментарію.

Таким чином, на прикладі неповного переліку існуючих технологій веброзробки стає зрозуміло, наскільки ця галузь технологій є розгалуженою, технологічною, потребує широкого набору компетенцій і навчання.

З розвитком вебпрограмування, закономірно збільшувалась і кількість вебсайтів з доволі різних тем та галузей. Довідники та їм подібні теж не виняток. Але якщо взяти за увагу матеріали для веброзробки та програмування саме спеціалізованих на українську аудиторію, то зрівнюючи їх кількість з зарубіжними, стає очевидно, що таких довідників саме для українських програмістів в наявності дуже мало. Нижче будуть розглянуті навчальні сайти-довідники, націлені саме на вебпрограмування.

### 1. HTML та CSS довідник українською [14] (рис. 1.2).

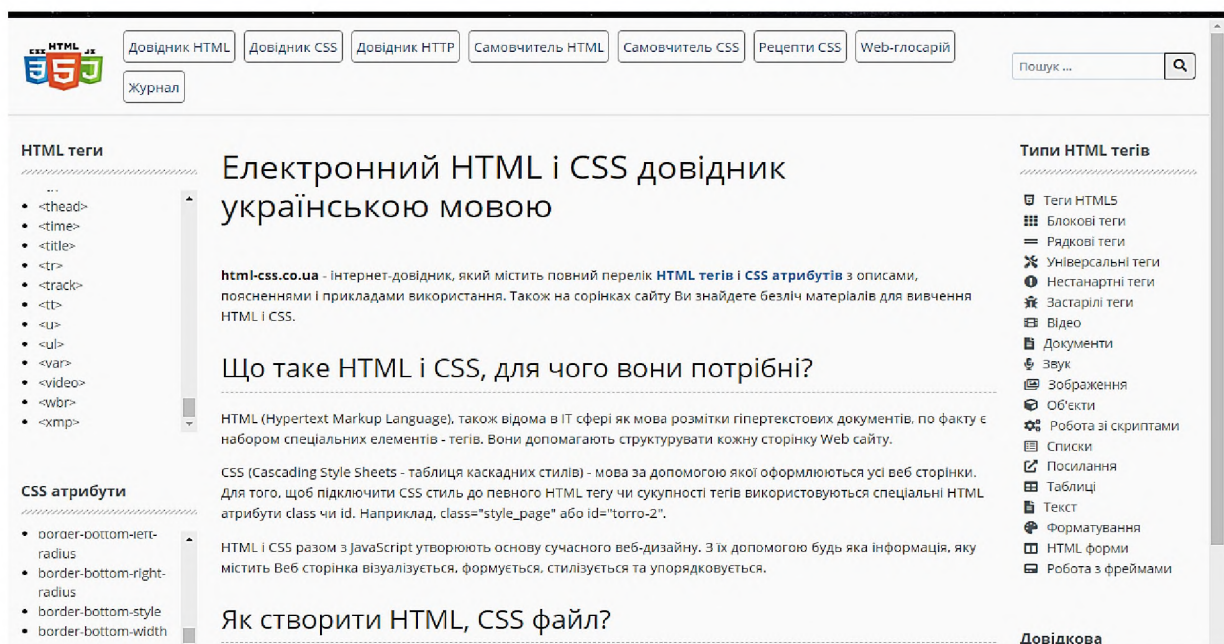


Рисунок 1.2- Вигляд сайту «Електронний HTML та CSS довідник українською»

Наведений україномовний сайт (див. рис. 1.2) містить навчальні матеріали з HTML та CSS, створений командою вебдизайнерів та розробників у 2017 р. Їхня мета була створити інформативний та зручний ресурс рідною мовою людям, які хочуть розширити знання у веброзробці. На сайті розміщено два великі розділи: HTML та CSS. Вони містять набір інструкцій, які описують як базові, так і більш просунуті теми мови. Підручники написані у простій та зрозумілій формі та підкріплені знімками екрану та кодовими прикладами.

На вебсайті, що містить безліч матеріалів з веброзробки на мовах HTML та CSS, є великі та зручні для користувача навчальні посібники з добре прописаним матеріалом [15]. Навігація та інтерфейс інтуїтивно зрозумілі. Ясний та лаконічний макет робить його простим у використанні, адаптивний дизайн дозволяє переглядати його на мобільних пристроях. Вебсайт [html-css.co.ua](http://html-css.co.ua) – чудовий довідник, перевагою якого є безкоштовне використання джерел інформації про вебзробку, передбачених українською мовою.

## 2. Український вебдовідник CSS.in.ua [16] (рис. 1.3).

The screenshot shows the website [CSS.in.ua](http://css.in.ua). The header features the site logo and a navigation menu with links for Home, CSS, HTML, JS, Articles, and News. The main content area is divided into sections for CSS and HTML. The CSS section includes a description of Cascading Style Sheets, a code example for styling a body and h1, and buttons for 'CSS properties' and 'CSS selectors'. The HTML section includes a description of HyperText Markup Language, a code example for basic HTML structure, and buttons for 'HTML tags', 'HTML elements', and 'HTML global attributes'. A sidebar on the left provides quick links to various guides and topics.

Рисунок 1.3 – Інформаційний розділ вебсайту CSS.in.ua

CSS.in.ua – це ще один український вебсайт, який надає довідкову інформацію з мови HTML та каскадних таблиць стилів (CSS), а JavaScript знаходиться в розробці. Сайт розділений на три основні розділи: властивості CSS, селектори CSS та розмітка HTML (див. рис. 1.3). Кожен розділ містить докладний список доступних опцій з прикладами та поясненнями.

У розділі властивостей CSS на сайті перелічені всі доступні властивості CSS, а також їх описи, значення та приклади. Атрибути організовані в категорії, такі як текст, фон, кордон та поля.

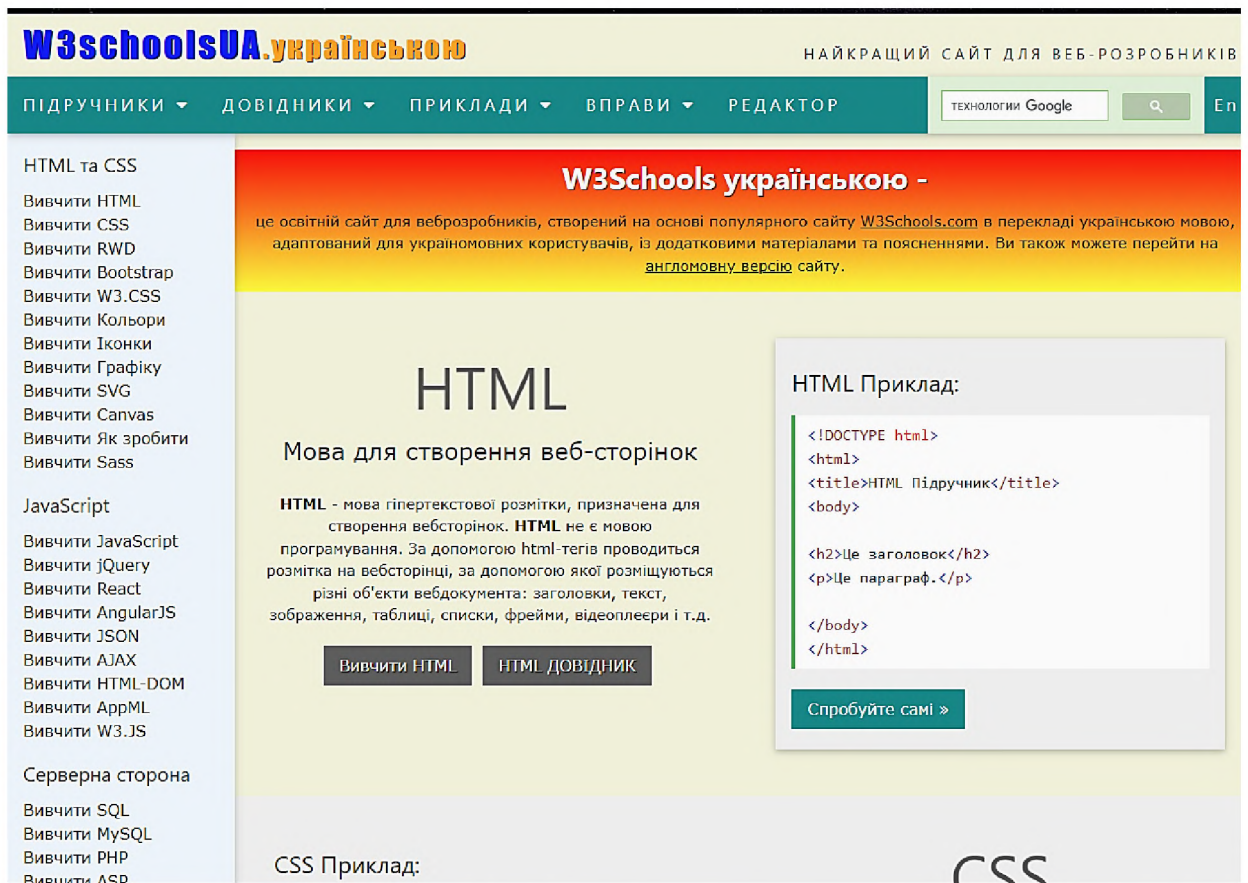
У розділі селекторів CSS на сайті перелічені всі доступні селектори CSS, а також їх описи та приклади. Селектори організовані за типом, наприклад, селектори елементів, селектори атрибутів та селектори класів.

У розділі HTML-теги сайту наведено всі доступні HTML-теги, а також їх описи та приклади. Вкладки організовані за категоріями, такими як заголовки, абзаци, списки та таблиці.

Крім посібників, CSS.in.ua містить велику кількість корисних статей про вебпрограмування, стосовно не лише моментів кодування, а й реалізації, теорії та різного виду плагінів для розробки сайтів.

CSS.in.ua – цінний ресурс для всіх, хто хоче більше дізнатися про вебпрограмування. Сайт зручний у використанні, містить актуальну інформацію. Доволі великою перевагою є також те, що його можна використовувати безкоштовно. Але, сайт був створений зовсім недавно, текстова структура вебсайту є недосконалою, матеріал подається не дуже влучно, часто трапляються помилки як в самому тексті, так і в прикладах коду. Зважаючи на недавнє створення ресурсу, це може бути зрозуміло. Ймовірно, з часом адміністратори сайту виправлять помилки і поліпшать якість матеріалу. Для вебпрограмістів, які шукають безкоштовні ресурси, CSS.in.ua може бути корисним додатком, проте рекомендується перевіряти інформацію з інших надійних джерел для підтвердження точності та уникнення неправильного використання коду.

3. HTML, CSS, JavaScript, PHP. W3Schools українською [17] (рис. 1.4).



**W3schoolsUA.українською** НАЙКРАЩИЙ САЙТ ДЛЯ ВЕБ-РОЗРОБНИКІВ

ПІДРУЧНИКИ ▾ ДОВІДНИКИ ▾ ПРИКЛАДИ ▾ ВПРАВИ ▾ РЕДАКТОР

**HTML та CSS**

- Вивчити HTML
- Вивчити CSS
- Вивчити RWD
- Вивчити Bootstrap
- Вивчити W3.CSS
- Вивчити Кольори
- Вивчити Іконки
- Вивчити Графіку
- Вивчити SVG
- Вивчити Canvas
- Вивчити Як зробити
- Вивчити Sass

**JavaScript**

- Вивчити JavaScript
- Вивчити jQuery
- Вивчити React
- Вивчити AngularJS
- Вивчити JSON
- Вивчити AJAX
- Вивчити HTML-DOM
- Вивчити AppML
- Вивчити W3.JS

**Серверна сторона**

- Вивчити SQL
- Вивчити MySQL
- Вивчити PHP
- Вивчити ASP

**W3Schools українською -**  
це освітній сайт для веброзробників, створений на основі популярного сайту W3Schools.com в перекладі українською мовою, адаптований для україномовних користувачів, із додатковими матеріалами та поясненнями. Ви також можете перейти на [англomовну версію сайту](#).

# HTML

## Мова для створення веб-сторінок

**HTML** - мова гіпертекстової розмітки, призначена для створення вебсторінок. **HTML** не є мовою програмування. За допомогою html-тегів проводиться розмітка на вебсторінці, за допомогою якої розміщуються різні об'єкти вебдокумента: заголовки, текст, зображення, таблиці, списки, фрейми, відеоплеєри і т.д.

[Вивчити HTML](#) [HTML ДОВІДНИК](#)

**HTML Приклад:**

```
<!DOCTYPE html>
<html>
<title>HTML Підручник</title>
<body>
<h2>Це заголовок</h2>
<p>Це параграф.</p>
</body>
</html>
```

[Спробуйте самі >](#)

**CSS Приклад:** CSS

Рисунок 1.4 – Вигляд сайту HTML, CSS, JavaScript, PHP. W3Schools

W3Schools – вебдовідник, який добре зарекомендував себе як гарний помічник, що пропонує довідкові матеріали та навчальні посібники з тем веброзробки (див. рис. 1.4). Широкий спектр тем вебсайту включає Python, SQL, JavaScript, CSS, PHP та HTML. Його структура гарно організована, та на додаток має велику кількість корисних функцій, таких як онлайн-редактори та програми сертифікації.

Той, хто зацікавлений у підвищенні своїх навичок у веброзробці, знайде цей довідник гарним ресурсом. Сайт пропонує високоякісний та актуальний контент для користувача, що лаконічно супроводжується яскравими прикладами.

Сучасний та чистий дизайн довідкового сайту є доволі привабливим. Пошук контенту не вимагає зусиль, а навігація сайтом проста. Функціонал сайту більш ніж достатній, задовольняє всі потреби по навігації та перегляду довідника.

Вміст вебсайту доволі є гарно наповненим. Він включає в себе посібники з різних аспектів по веброзробці, питання, що часто ставляться, інформаційні статті, має навчальні відеоролики, покрокові інструкції та довідники користувача для покращення функціональності як самого сайту, так і зворотного зв'язку.

Маючи гарно подану та цінну інформацію для тих, хто працює чи планує працювати в сфері вебпрограмування, W3School є гарним і актуальним ресурсом. Заслуговує на увагу також його повна безкоштовність. На тлі всіх українських вебдовідників його можна вважати найкращим та найбільш повним зі сторони наявності інформації стосовно всіх можливих мов веброзробки.

#### 4. HubSpot, The Beginner's Guide to Website Development [17] (рис. 1.5).

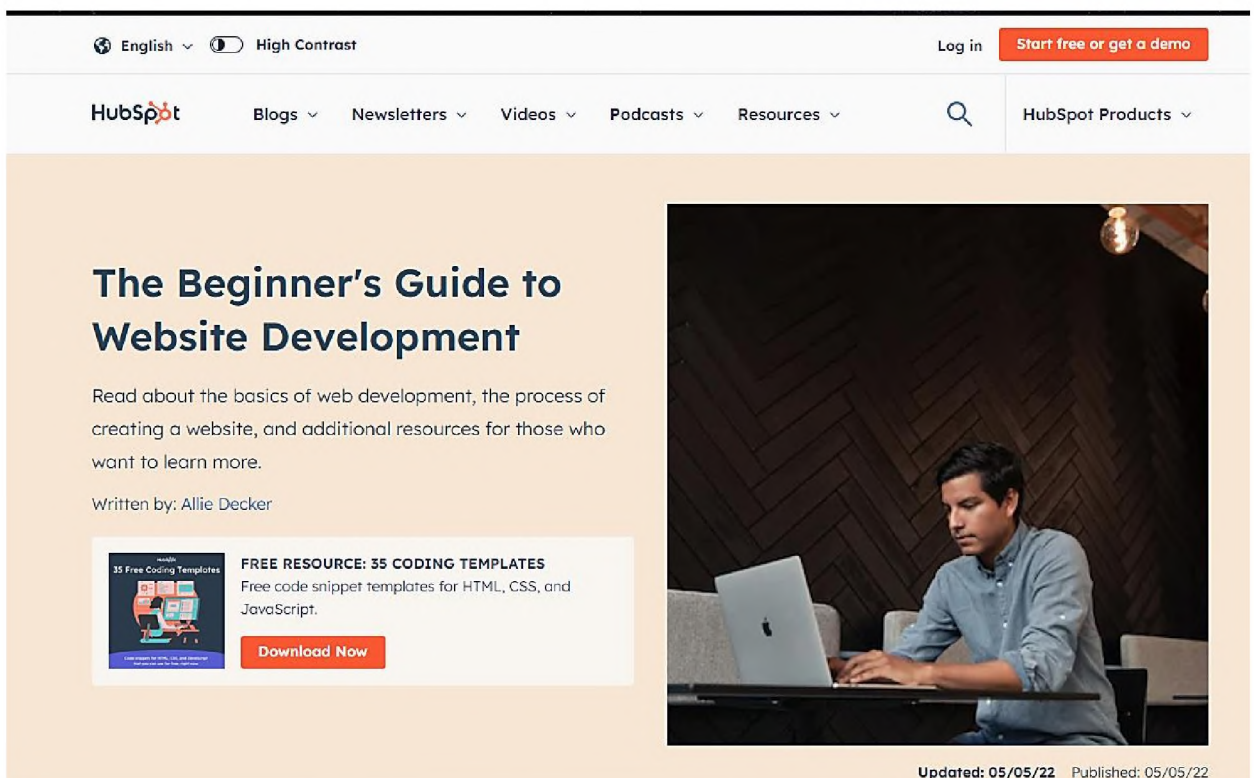


Рисунок 1.5 – Вигляд сайту HubSpot

Удосконалення чи створення вебсайту стає простішим завдяки блогу HubSpot (див. рис. 1.5), призначеному як для компаній будь-якого розміру, так і для поодиноких веброзробників. Такі теми, як безпека, маркетинг,

дизайн вебсайту та розробка, висвітлюються в цьому блозі розробки вебсайту. Авторами блогу команда веброзробників і маркетологів, які вміло подали містку чітку та стислу інформацію на своєму ресурсі. Він надає розробникам корисні ресурси, зокрема шаблони, тематичні дослідження та навчальні посібники для покращення їхніх вебсайтів. Інформативний зміст є основною частиною цього блогу, де регулярно публікується нова інформація, виходячи з оновлень на просторах веброзробки. Ці дописи в блозі надають розробникам цінну та актуальну інформацію, яку можна використовувати для покращення вже створених вебсайтів. На блозі висвітлені матеріали не тільки з веброзробки а і з різними нюансами впровадження, реалізації, підбору дизайнів, маркетингу тощо. Сама матеріальна база є доволі наповненою.

Блог розробки вебсайтів HubSpot є прекрасним ресурсом для всіх, без винятків по досвіду, розробників, які прагнуть створити чи покращити вже створений сайт [18]. У блозі представлена чітка та розгорнута інформація, яка добре сприймається користувачем. Багато матеріалу та ресурсів викладено безкоштовно, але якщо користувачу потрібно більш детально вивчити матеріали по веброзробці чи зв'язаною з нею інформацією, то йому буде необхідно буде отримати платний доступ.

Виходячи із аналізу наявних ресурсів, можна зробити наочний висновок, що для української аудиторії довідників з веброзробки доволі мало, не всі зроблені досконало, в порівнянні з зарубіжними. Тому такого контексту довідники являються актуальними та затребуваними.

### **1.3 Характеристика типів існуючих вебсайтів інформаційно-довідкового характеру (Портали, відкриті бібліотеки, підручники тощо)**

З появою всесвітньої павутини інтернет, життя для багатьох користувачів спростилося. Адже з плином часу, в мережі почали з'являтися різного роду інформаційно-довідникові вебсайти. Будь-якого дня людям

потрібен доступ до інформації з різних галузей чи життєвих питань. Таким чином, інформаційні та довідкові вебсайти є життєва важливими, оскільки вони дозволяють людям отримувати необхідну інформацію в будь-який час і в будь-якому місці. Користувачі можуть знайти саме те, що їм потрібне, не виходячи з дому, лишень зробивши запит на своєму смартфоні чи ПК.

Існує кілька типів вебсайтів інформаційно-довідкового характеру, які відрізняються за своєю специфікою та призначенням:

1. Портали – це великомасштабні вебсайти, які поєднують різні інформаційні та службові ресурси, такі як новини, погода, фінанси, розваги, освіта та лікування [19]. Портали зазвичай надають своїм користувачам персональний контент, мають потужні пошукові системи, форуми, блоги, чати та інші інструменти для спілкування та взаємодії. Яскравим прикладом порталу є український вебсайт Ukr.net [20].

2. Відкриті бібліотеки – це вебсайти, які надають безкоштовний доступ до цифрових книг, статей, дослідницьких робіт, рефератів та інших матеріалів [21]. Відкриті бібліотеки існують у всіх галузях знань, від літератури та мистецтва до науки та техніки. Прикладом відкритої бібліотеки можна вважати електронну бібліотеку «Культура Україна» [22].

3. Підручники – це вебсайти, які надають безкоштовний доступ до навчальних матеріалів, таких як лекції, завдання, тести тощо. Підручники можуть бути орієнтовані на різні рівні освіти, від початкової до вищої [19].

4. Енциклопедії – це вебсайти, які надають широкий спектр знань та інформації з різних предметів, тем та концепцій [21]. Енциклопедії можуть містити статті, ілюстрації, графіку та інші форми інформації. Найбільш яскравим прикладом енциклопедії можна назвати вебсайт «Вікіпедія» [23].

5. Довідники – це вебсайти, які надають інформацію з певної теми, такої як медицина, фінанси, юриспруденція, бізнес, спорт тощо. Каталоги можуть включати статті, описи, поради, огляди, рейтинги та іншу інформацію. Як приклад, можна взяти один із найбільш популярних україномовних підручників «Pidru4niki, навчальні матеріали онлайн» [24].

6. Форуми – це вебсайти, які надають платформу для обговорення різних тем, ідей, запитань та відповідей [19]. На форумах, зазвичай, є розділи різних тем, і користувачі можуть створювати нові теми для обговорення. Форуми можна відкривати та закривати для реєстрації нових користувачів. Яскравим прикладом форуму можна назвати вебсайт «DOU. Спільнота програмістів» [25].

7. Блоги – це вебсайти, які надають користувачам можливість створювати та публікувати контент у вигляді статей, нотаток, думок тощо. Блоги часто мають особистий характер, тому вони дозволяють читачам краще зрозуміти автора та його погляди на різні теми. Гарним прикладом блогу може виступити вебсайт «KITWeb» [26].

Інформативні та довідкові вебсайти можуть служити різним цілям, включаючи освіту, самоосвіту, маркетинг та комерцію тощо. Створення вебсайту передбачає вибір типу довідкового чи інформаційного сайту, який найкраще відповідає цілям та використанню аудиторії. Кожен тип має певні переваги та характеристики, які слід враховувати під час ухвалення рішення.

Отже виходячи з даної інформації, можна сказати, що подібні довідникові вебсайти та матеріали, що розміщені в мережі інтернет зараз є невід’ємною частиною життя майже будь-якого користувача мережі інтернет. Саме так й розроблюваний вебсайт має на меті допомогти в навчанні, роботі та покращенні своїх знань будь-якому українському веброзробнику, незалежно від його досвіду.

## РОЗДІЛ 2

### ОБҐРУНТУВАННЯ ПЛАНУ РОЗРОБКИ ВЕБДОДАТКУ ДЛЯ НЕФОРМАЛЬНОГО НАВЧАННЯ ТА ВИБОРУ ТЕХНОЛОГІЙ

#### 2.1 Опис вимог до проєкту навчального вебдодатку на основі дослідження аналогів

Під час створення будь-якого вебдодатку, потрібно визначати основні критерії, які будуть присутні в розрізі предметної області, тобто ті матеріали, функції чи інші аспекти, які повинні бути присутні на сайті того чи іншого типу, задля максимальної привабливості та правильності розроблюваного вебдодатку. Такі додатки повинні мати в собі перелік потрібних компонентів, функцій та контенту для максимально вдалого використання користувачем.

Завданням кваліфікаційної роботи є розробка навчального вебдодатку для неформальної освіти. Враховуючи популярність вивчення іноземних мов, прийнято рішення спроектувати інтерактивний вебсайт, приміром, для вивчення іноземної мови. Для початку проведено запити щодо наявних вебсайтів аналогічного призначення та проведено всебічний аналіз таких додатків. За запитом в Google було знайдено більше 50 варіантів пропозицій в Україні, зокрема в Полтаві. Зібрали для огляду такі, що мають різні сервіси та дизайн. Для формування критеріїв аналізу виділили перелік особливо важливих пунктів, які необхідно врахувати. Їхній опис наведено далі.

1. Дизайн та навігація: сайт повинен мати сучасний та привабливий дизайн [27], який повинен бути більш спрощеним, без непотрібних компонентів, які б відволікали від основного контенту. Головне меню та вся інша навігація повинна бути простою та легкодоступною для користувача, знаходитися в зручних місцях, не перекриваючи основний контент [19]. Для контенту повинен бути виділено достатньо простору для зручності його сприйняття. На всіх сторінках сайту повинен використовуватися єдиний стиль будови: зовнішній вигляд структури сайту повинен залишитись однаковим.

2. Зміст: вебдодаток має пропонувати достатню базу знань, що охоплює основну тематику, представлену у різних форматах. Тут можуть бути присутні посібники, інструкції, поради, приклади та інші цінні матеріали [27]. Мета вебдодатку - надати користувачам корисну, точну та актуальну інформацію для допомоги, навчання і розширення своїх знань та досягненні успіху в своєму бізнесі, освіті.

3. Категоризація та структура: сайт повинен бути організованим за категоріями або тематичними розділами, що дозволяє користувачам легко орієнтуватися та знаходити інформацію, яка їх цікавить [27]. Категорії можуть мати розділи та підрозділи, тобто сама структура довідника має бути ієрархічно організованою [28]. Контент повинен логічно розділятися по тематиці контенту, подавати надалі інформацію пов'язану з вже обраною тематикою [29]. Правильна конфігурація вебсайту дозволить користувачам ефективно переміщатися по сегментах, легко знаходити потрібні ресурси і не втрачати орієнтацію в морі даних, що надаються.

4. Оновлення та підтримка. Вебдодаток повинен регулярно оновлюватися, щоб надавати найсвіжішу інформацію в розрізі свого контексту. На сайті може бути передбачена можливість залишати відгуки та пропозиції, щоб користувачі могли ділитися своєю думкою та пропонувати ідеї для покращення змісту, чи функціональності сайту. Регулярні оновлення допоможуть підтримувати інтерес користувачів та підвищать затребуваність розробленим вебдодатком.

5. Підтримка мультимедійних елементів: крім текстового контенту сайт може підтримувати різні формати мультимедіа [30]. Це збагатить інформацію та ефективно проілюструє контент, допоможе краще передати користувачеві матеріал та зможе задовільнити різні потреби для цілісної аудиторії.

Загальний аналіз продемонстрував актуальність обраного напрямку та існування найрізноманітніших прийомів втілення ідеї. Результати порівняльних характеристик обраних для порівняння вебдодатків із вивчення іноземних мов наведено в табл. 2.1.

Таблиця 2.1 – Порівняльні характеристики дизайну та функцій вебдодатків для надання послуг/інформації з вивчення іноземних мов

Критерії	Educado [31]	ABC Language School [32]	Speak It [33]	Svitlo School [34]
Класифікація вебдодатку	Промо-сайт і навчальні матеріали в хмарі	Інформаційний сайт школи іноземних мов	Промо-сайт	Інформаційний вебсайт школи
Тип вебдодатку	SPA (Single Page Application)	Класичний інформаційний вебсайт, багатосторінковий	SPA (Single Page Application)	Класичний інформаційний вебсайт, багатосторінковий
Стиль	Мінімалізм	Метро, блочна структура	Метро, блочна структура	Метро, блочна структура
Графіка	Айдентика, фото	Фото, фон, 2 базових кольори, акцент, фото	Фото, 2 базових кольори, акцент, фото, ікони	Фото, 2 базових кольори, акцент, фото, ікони
Логотип	+	+	+	+
Кількість розділів	3 частини	6 сторінок (меню)	10 сторінок	23 сторінки
Наявність форм	Для реєстрації	Відсутні	Запис на пробний урок, реєстрація	Реєстрація
AI асистент	Чат-бот	Немає	Немає	Немає
Канали комунікації	Соціальні мережі, телеграм-канал, замовлення зворотного дзвінка, телефони	Соціальні мережі, телеграм-канал, реєстрація на пробний урок, адреси офісів, телефони	Групи в Instagram, месенджери, реєстрація на пробний урок, адреси офісів, телефони, карта	Соціальні мережі, телеграм-канал, реєстрація на пробний урок, телефони, пошта, розсилка новин
Спеціальні динамічні ефекти	«Акордеон», кнопки переходу на форми	Слайдер, кнопки включення форми, gif-анімація	Слайдери, трансформації зображень, відео, зміна кольору при наведенні на об'єкт	Відсутні. фотогалереї
Засоби навігації	Меню, кнопки, посилання	Меню, кнопки, прокрутка сторінки мишею	Меню, стрілки, кнопки, блоки-переходи	
Адаптивність	+	+	+	+

Вебдодатки, які описані в табл. 2.1, мають спільне завдання: надати інформацію про можливості вивчення іноземних мов (однієї або кількох), місце і умови отримання послуг, відгуки користувачів та інші елементи. Щодо

дизайну, структури, технологій реалізації, то вони є доволі різними. Наприклад, онлайн-школа Educado має оригінальний мінімалістичний дизайн у формі SPA (рис. 2.1), ABC LanguageSchool – класичний багатосторінковий (рис. 2.2). При цьому надаються послуги з вивчення багатьох мов.

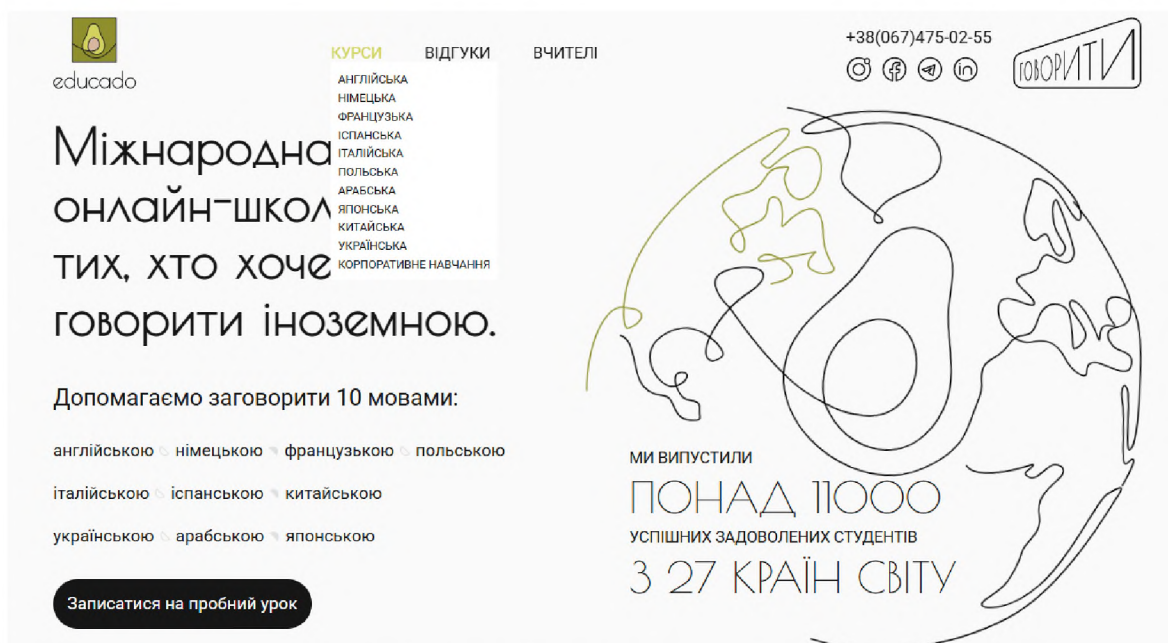


Рисунок 2.1 – Головна сторінка школи Educado

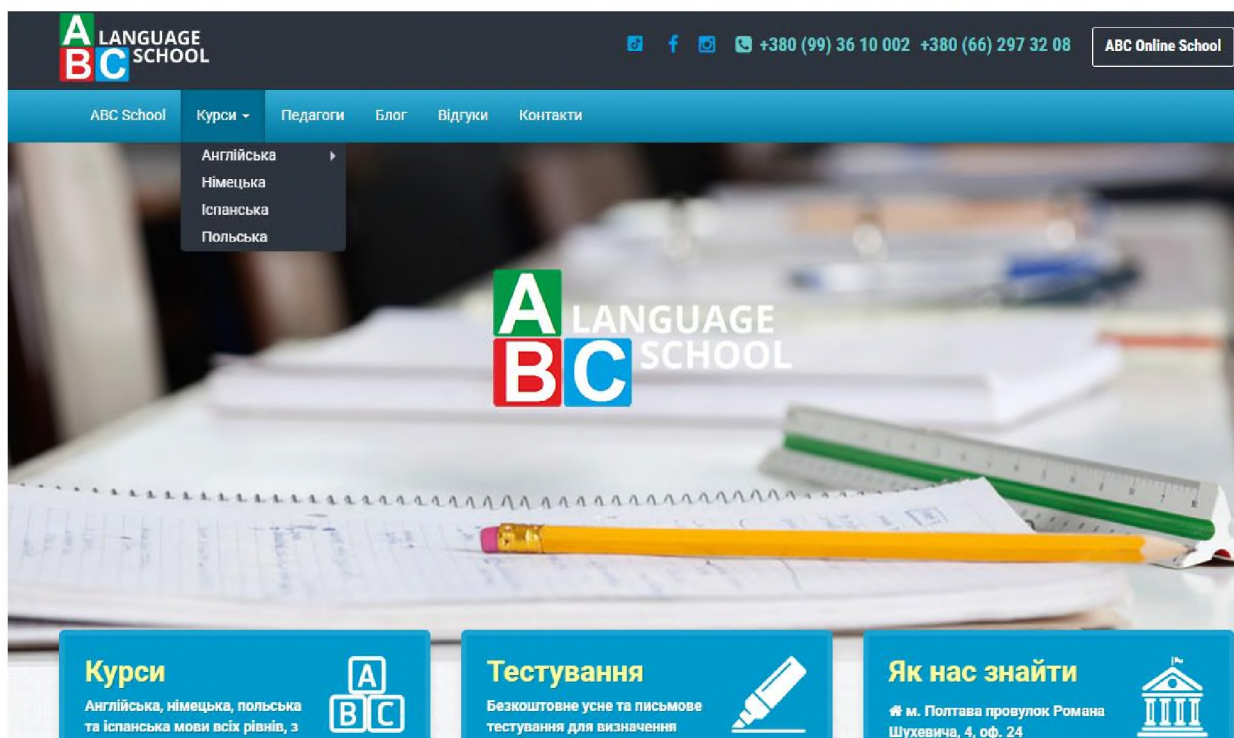


Рисунок 2.2 – Структура головного меню вебдодатку ABC LanguageSchool

Узагальнюючи проведені дослідження, були виділені ключові вимоги до вебдодатків з вивчення іноземних мов, щоб бути ефективними, зручними та мотивуючими для користувачів. Далі опис ключових із них.

#### 1. Інтерактивність може включати:

- динамічні вправи: тести, перетягування слів, вправи на відповідність, диктанти, усні відповіді;
- миттєвий зворотний зв'язок: користувач має одразу бачити свої помилки та отримувати пояснення;
- гейміфікація: бали, рівні, досягнення, які стимулюють до навчання.

#### 2. Адаптивність і персоналізація:

- рівневий підхід: контент повинен відповідати рівню володіння мовою (від A1 до C2);
- аналіз прогресу: система має відстежувати результати й підлаштовувати завдання;
- індивідуальні маршрути: користувач отримує завдання, які відповідають його сильним і слабким сторонам.

#### 3. Зручність користування (Usability)

- інтуїтивний інтерфейс: простий, зрозумілий дизайн навіть для користувачів без технічного досвіду;
- адаптація під різні пристрої: кросплатформеність (комп'ютери, планшети, смартфони);
- швидка навігація та зручна структура уроків.

#### 4. Підтримка мовного середовища:

- інтеракція з іншими: можливість спілкуватися з викладачами або іншими учнями (чати, форуми, відеозустрічі);
- симуляції реальних ситуацій: рольові ігри, віртуальні діалоги, вправи з побутової лексики.

#### 5. Можливість офлайн-доступу (бажано):

- навіть частковий доступ до матеріалів без підключення до Інтернету може бути перевагою.

6. Підтримка кількох мов інтерфейсу: це особливо важливо для початківців, яким складно розібратися в інтерфейсі іноземною мовою.

Відповідність цим вимогам підвищує ефективність навчання, покращує користувацький досвід і сприяє довгостроковому залученню до освітнього процесу. Для реалізації обраних функцій були розглянуті та обрані відповідні технології.

## 2.2 Обґрунтування та опис обраних технологій

Технології можна розділити за видом забезпечення функцій вебдодатку.

1. Frontend-технології (інтерфейс користувача) можуть включати:

– базові технології для створення структури, стилю та інтерактивності вебсторінки: HTML5 / CSS3 / JavaScript.

– React, Vue або Angular - сучасні JavaScript-фреймворки для побудови динамічних, швидких і компонентних інтерфейсів. Наприклад, React дозволяє створювати інтерактивні вправи та оновлювати сторінку без перезавантаження.

– Tailwind CSS або Bootstrap - фреймворки для швидкої верстки й адаптивного дизайну. Зручно для кросплатформенності.

– для роботи з аудіо-прослуховування, запис голосу, розпізнавання мови, переведення текст-у-мову: Web Audio API / Web Speech API.

HTML (HyperText Markup Language) та CSS (Cascading Style Sheets) є основними мовними технологіями для створення вебсайтів та визначення їхнього зовнішнього вигляду та являються базою вебпрограмування [35]. Вони залишаються актуальними та важливими інструментами у веброзробці як сьогодні, так, швидше всього, будуть у використанні в найближчому майбутньому. З цими технологіями зустрічаються як нові розробники, так і працюють більш досвідчені. Ці мови є доволі простими, не потребують

додаткових знань з інших мов програмування, та знову ж таки являються ключовими та базовими мовами у веброзробці [2].

HTML – це мова розмітки, яка використовує теги для визначення різних елементів вебсторінки [36]. HTML використовується для створення структури вебсторінки, тоді як CSS використовується для її стилізації, тобто CSS – це мова таблиць стилів, яка використовує правила для керування зовнішнім виглядом елементів HTML (див. рис. 2.2) [37]. Разом вони дозволяють веброзробникам створювати візуально привабливі та зручні вебсайти.

Інші технології можна комбінувати з HTML та CSS, що дозволяє створювати більш потужні та складніші вебдодатки. Ці сторонні технології включають системи керування контентом (CMS), платформи JavaScript та різні бібліотеки і фреймворки.

При розробці інтерактивних вебдодатків важливо продумати та зважити ступінь застосування елементів Backend-технологій (логіка, база даних, API). Серед найбільш популярних використовують наступні:

- Node.js / Express.js - популярна серверна платформа для обробки запитів, збереження результатів тестів, керування користувачами;
- Python (Django або Flask) - альтернатива для серверної частини, зручно при інтеграції з AI/ML;
- Бази даних (PostgreSQL, MongoDB, Firebase) - для зберігання інформації про користувачів, прогрес, уроки та матеріали;
- REST API або GraphQL - системи обміну даними між клієнтом і сервером; REST – стандартний варіант, GraphQL – більш гнучкий.

Серед вказаних технологій достатньо засобів, які можуть забезпечити задані параметри вебдодатку для представлення інформації, інтерактивного управління елементами інтерфейсу з привертанням уваги до елементів меню, блочного контенту через анімації, трансформації, та ін., тобто, забезпечують базову структуру і функції. При створенні повноцінного додатку для онлайн навчання потрібні окремі й додаткові засоби, перелік і функції яких представлено в табл. 2.2.

Таблиця 2.2 – Набір додаткових технологій і сервісів для забезпечення розробки повнофункціонального вебдодатку для онлайн-навчання

Напрямок забезпечення	Перелік технологій	Результат застосування
Хмарні сервіси та інтеграції	Firebase або Supabase	Платформи для автентифікації користувачів, зберігання даних, хостингу і push-нотифікацій
	AWS / Google Cloud / Azure	Хостинг, масштабованість, аналітика, хмарні обчислення (наприклад, для AI-функцій або зберігання відео/аудіо).
	Content Delivery Network (CDN)	Швидка доставка медіаконтенту в різних регіонах
Штучний інтелект і машинне навчання	ChatGPT API / OpenAI API / Google PaLM	Генерація діалогів, автоматичне виправлення помилок, створення персоналізованих рекомендацій.
	Speech-to-Text і Text-to-Speech (Google, Amazon, Microsoft)	Для роботи з усним мовленням: розпізнавання та синтез голосу
	Recommendation Systems	Аналіз поведінки користувача та пропозиція відповідного контенту
Системи управління контентом (CMS)	Strapi, Sanity або власна CMS	Для адміністраторів: створення й редагування уроків, тестів, слів і вправ без необхідності змінювати код
Тестування та аналітика	Jest / Cypress	Інструменти для автоматичного тестування функціональності додатку
	Google Analytics / Hotjar	Збір статистики, аналіз поведінки користувачів, покращення UX/UI.
Додаткові інтеграції	Платіжні системи: Stripe, LiqPay або PayPal	Для проведення операцій оплати платних курсів
	Інтеграція з соцмережами	Реєстрація, шерінг результатів
	Zoom/WebRTC	Для живих занять або мовної практики.

Перелічені вище технології (у т. ч. табл. 2.2) разом створюють потужну систему, яка дозволяє розробити масштабований, адаптивний та інтерактивний вебдодаток, орієнтований на ефективне навчання іноземним мовам. Проаналізувавши наявні в мережі застосунки та можливості сучасного веброзробника, можна остаточно сформулювати кінцеві цілі в рамках реалізації власного проєкту та обрати з наведеного переліку оптимальний стек технологій, які дозволять розробити вебдодаток з вивчення однієї іноземної мови. Він повинен містити вибір рівнів навчання, рекламу школи, бажано у вигляді слайдера, форму реєстрації, зручне пересування між навчальними розділами.

Оптимальний стек запланованих до використання при розробці вебдодатку технологій виглядає наступним чином.

Для клієнтської частини обираємо основні та додаткові:

- інструменти HTML та CSS, за допомогою яких планується налаштувати зовнішній вигляд вебсторінки та покращити взаємодію з користувачем;
- використання React – компонентний фреймворк, що ідеально підходить для побудови динамічних інтерфейсів, у тому числі слайдерів, перемикачів рівнів, переходів між розділами тощо;
- Tailwind CSS – сучасна утилітна CSS-бібліотека для швидкого, адаптивного, чистого дизайну без потреби писати багато кастомного CSS;
- React Router – для зручної навігації між розділами додатку (уроки, головна, реєстрація, про школу тощо);
- Swiper.js або React Slick – бібліотека для реалізації слайдера реклами школи, а також навчальних слайдів, якщо потрібно.

Для елементів Backend-частини (серверної логіки) можна обрати швидкий та надійний серверний фреймворк Node.js/Express.js, який добре підходить для обробки запитів, аутентифікації користувачів, збереження прогресу навчання, управління уроками тощо.

Плануються застосування й інших корисних інструментів:

- Formik + Yup – для створення та валідації форм (наприклад, форма реєстрації);
- Framer Motion – для анімацій переходів між сторінками та елементами;
- React Hook Form – альтернатива для роботи з формами;
- React Helmet – для SEO-оптимізації сторінок (наприклад, опис школи).

Обрання зазначеного стеку технологій, окрім простоти і швидкості розробки, забезпечить: масштабованість у майбутньому (можна додати чат, гейміфікацію, інтеграцію зі штучним інтелектом); гнучкий дизайн і адаптацію

до мобільних пристроїв тощо. Такий набір технологій підходить для як односторінкового застосунку (SPA), який на початку представляє даний проєкт, так і багатосторінкового.

### 2.3 Планування структури вебдодатку, створення основного макету та робота з кодом

Враховуючи опис вимог до проєкту вебдодатку і заданих функцій, наведених у попередніх підрозділах, вирішено створити односторінковий вебдодаток (лендінг, SPA). Макет головної сторінки розроблено в середовищі Figma [38]. Фрагмент процесу проєктування макету показано на рис. 2.3.

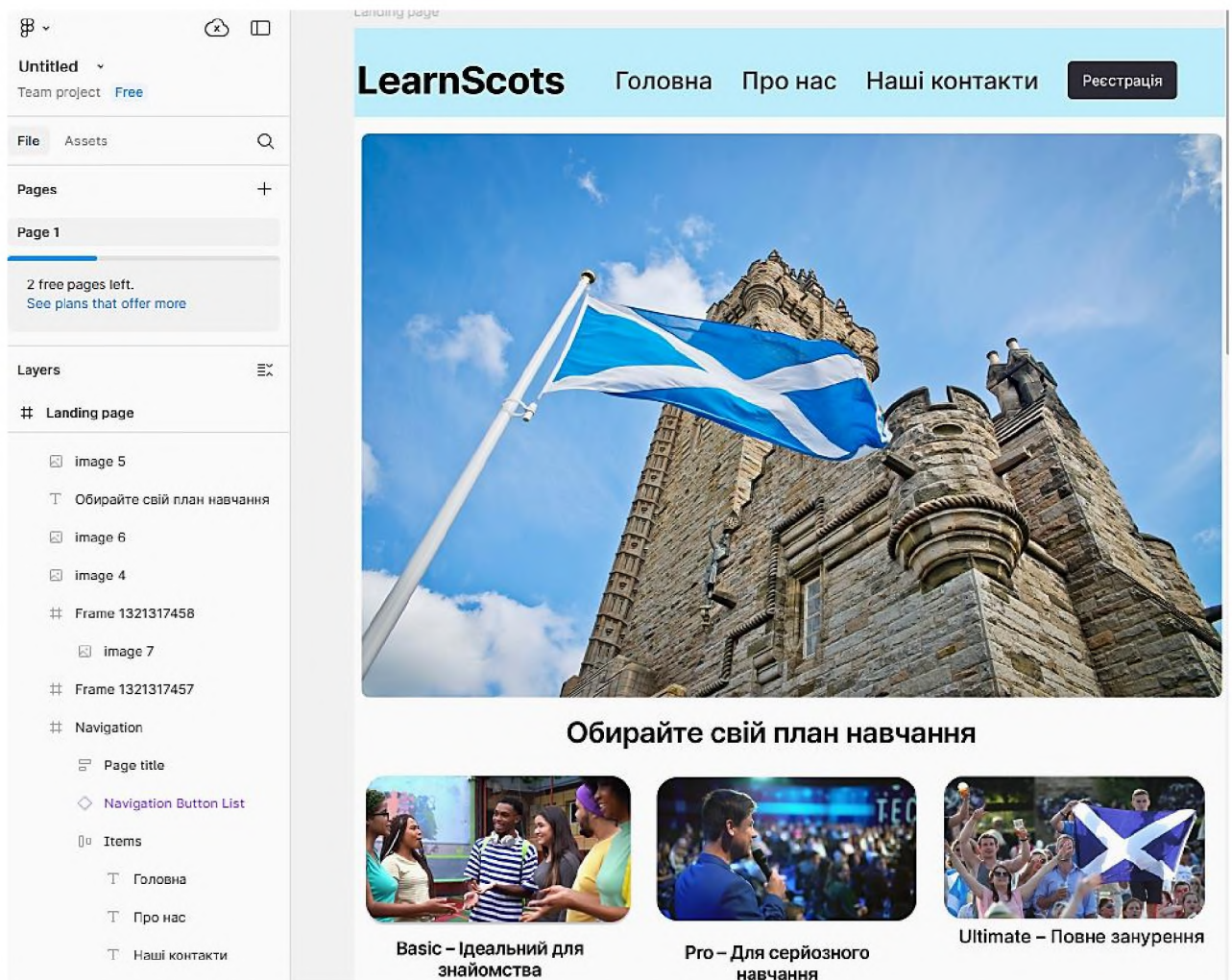


Рисунок 2.3 – Проєктування у Figma макету головної сторінки вебдодатку

Планування структури сайту – це процес визначення організації та ієрархії інформаційних елементів, сторінок та розділів на вебсайті. Він включає розробку логічної структури, навігації та зв'язків між різними компонентами вебсайту.

Редактор Figma – це потужний інструмент для створення макетів вебсторінок, який має низку переваг. Він працює прямо в браузері, що дозволяє легко співпрацювати в команді в реальному часі. Інтерфейс інтуїтивно зрозумілий, а велика кількість готових компонентів і плагінів значно пришвидшує розробку. Крім того, Figma підтримує адаптивний дизайн, спрощує передачу макетів розробникам і забезпечує точність відображення елементів на різних екранах.

Формування коду вебдодатку розпочато з використання HTML і CSS. HTML визначає структуру та організацію таких елементів, як заголовки, абзаци, списки, таблиці тощо [39]. Приклад верстки верхньої частини («хедер») вебдодатку як HTML-код у редакторі Visual Studio Code (VSC), показано на рис. 2.4.

```

19 <body class="bg-gray-100">
20   <!-- НАВИГАЦІЯ -->
21   <header class="bg-white shadow-md fixed top-0 left-0 w-full z-50">
22     <nav class="container mx-auto flex justify-between items-center p-4">
23       <h1 class="text-2xl font-bold">LearnScots</h1>
24       <ul class="flex space-x-6">
25         <li>
26           <a href="#top" >Головна</a>
27         </li>
28         <li>
29           <a href="#why-choose-us"> Про нас </a>
30         </li>
31         <li>
32           <a href="#footer"> Контакти </a>
33         </li>
34       </ul>
35     </nav>
36   </header>

```

Рисунок 2.4 – Просктування верхнього розділу головної сторінки на HTML

Як видно (див. рис. 2.4), на початку роботи використано семантичні теги, наприклад, </header>, </nav> та ін. Наявність семантичних тегів дозволяє

скоротити і раціоналізувати код, а також працювати над проектом у групі, забезпечити пошукову оптимізацію. Одразу додані відповідні класи CSS, які будуть прописані окремо. Для розміщення блоків основної частини використані відповідно блочні елементи `</section>`, `<div>`, `<footer>` та інші.

Для досягнення відповідного дизайну та у відповідності до змісту були підібрані графічні елементи в інтернеті, які дозволені для використання на інших вебсайтах. Оскільки тематика вебдодатку – вивчення шотландської мови, то відповідно використана атрибутика країни, наприклад, на головному банері (рис. 2.5).

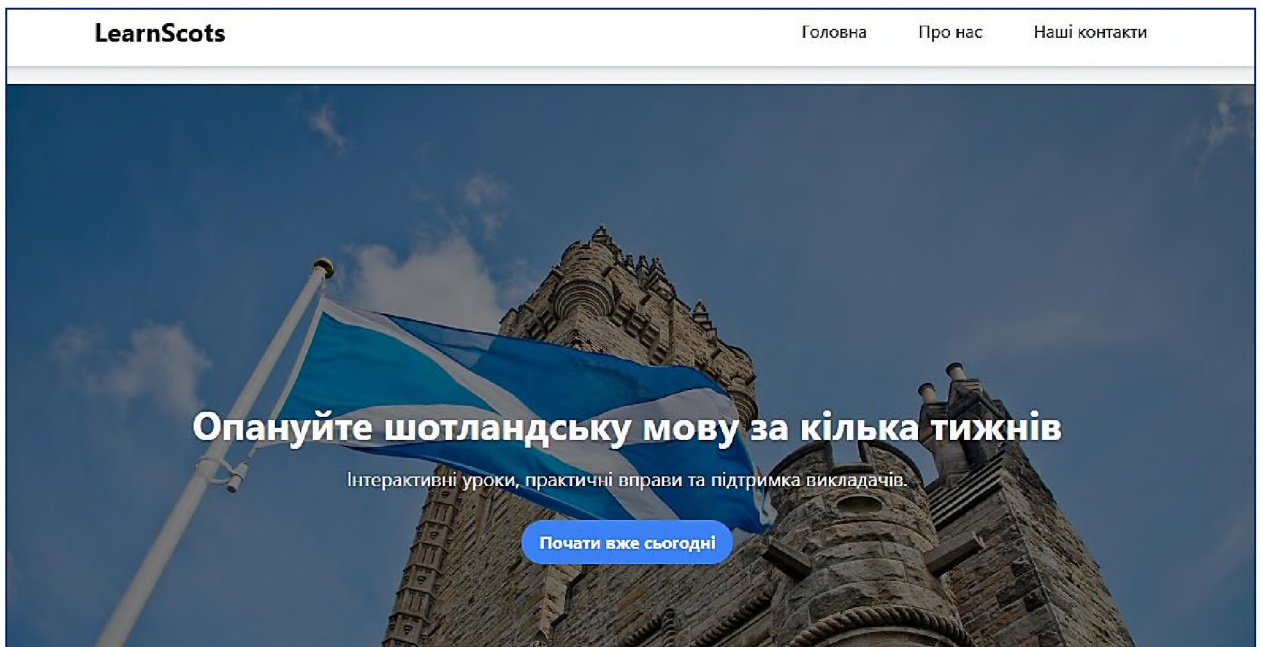


Рисунок 2.5 – Вигляд верхньої секції вебдодатку з тематичним банером та кнопкою переходу до розділів

Інтеграція з іншими технологіями: HTML та CSS тісно інтегровані з іншими технологіями веброзробки, такими як JavaScript. JavaScript дозволяє додавати динамічну поведінку та інтерактивність вебсторінок [40]. Його можна використовувати для обробки подій, створення анімації, відправлення та отримання даних із сервера та багато іншого. HTML і CSS у поєднанні з JavaScript відкривають широкі можливості для розробки вебдодатків та інтерактивних інтерфейсів користувача. Код банера показано на рис. 2.6.

```

29 <!-- БАНЕР -->
30 <section class="banner">
31   <div class="banner-overlay"></div>
32   <div class="banner-content">
33     <h2 class="text-4xl font-bold mb-4">Опануйте шотландську мову за кілька тижнів</h2>
34     <p class="text-lg mb-6">Інтерактивні уроки, практичні вправи та підтримка викладачів.</p>
35     <button onclick="document.getElementById('plans').scrollIntoView({ behavior: 'smooth' });"
36       class="btn-primary">Почати вже сьогодні</button>
37   </div>
38 </section>

```

Рисунок 2.6 – Код секції банера у верхній частині вебдодатку

На прикладі коду (див. рис. 2.6) можна побачити, що він містить лише розмітку та посилання на класи із описом всіх елементів стилю.

Повний HTML-код лендінгу наведено в додатку Б. Загальна структура проєкту показана схематично на рис. 2.7.



Рисунок 2.7 – Схематична структура проєкту і вкладені папка із зображеннями

На першому етапі верстки досягнуто таких результатів:

- адаптивний і привабливий дизайн;
- smooth scroll;
- секції з перевагами, відео, тарифами, футером;
- карусель відгуків;
- модульна структура з Tailwind'ом та Alpine.js.

Загалом HTML та CSS залишаються актуальними мовами та необхідними навичками для розробників вебсайтів і вебдодатків. Вони

надають базовий набір інструментів та інтегруються з іншими технологіями для створення більш складних та інтерактивних вебдодатків.

Планування структури вебсайту є одним із ключових аспектів під час розробки вебсайту, оскільки воно визначає, як користувачі будуть взаємодіяти з вашим вебсайтом, як вони знайдуть необхідну інформацію, використовуючи загальну навігацію [27]. Ефективна структура сайту спрощує пошук контенту та доступ до нього, підвищує зручність використання та покращує спільну взаємодію з користувачем.

Під час проектування потрібно переконатися, що кожна категорія є незалежною і не перетинається з іншими. Це покращить вид самого інформаційного наповнення на сторінці та полегшить прийняття інформації користувачів.

Організоване головне меню повинно бути простим та інтуїтивно зрозумілим для користувачів, а категорії мають бути легко доступні з будь-якої сторінки посібника, див. рис. 2.3. Посилання на підкатегорії та популярні розділи знаходяться в максимально зручному та доступному доступі на боковій панелі зліва (рис. 2.4).

Макет створювався на базі застосунку Visual Studio Code. Visual Studio Code (VS Code) – це текстовий редактор у вільному доступі, розроблений компанією Microsoft. Він призначений для розробки різних типів програмного забезпечення, включаючи вебпрограми, мобільні програми та хмарні сервіси. Для створення застосунку був обраний за свою зручність, обширну функціональність у розробці зокрема вебсайтів та простоту у використанні. Навігація та основний функціонал виповнений просто та влучно, що допомагає навіть недосвідченому розробнику швидко розібратися у його можливостях, вигляд застосунку та функціоналу зображено на рис. 2.5.

Анімовані елементи не акцентують на собі увагу, та мають за мету лише доповнити функціональну зручність додатку під час роботи з ним. Всі елементи навігації розміщені у видному та доступному для користувача вигляді.

Виходячи з приведеної інформації можна зробити висновок, що структура вебсайту є важливим етапом перед реалізацією коду. Під час розпланування структури необхідно визначити, як будуть організовані різні сторінки, розділи та підсторінки вебсайту. Це допомагає зрозуміти, як будуть пов'язані між собою різні елементи та як користувачі будуть переміщатися по вебсайту.

Таким чином, структура та макет вебдодатку взаємодіють між собою, допомагаючи розробникам та дизайнерам визначити організацію сторінок і елементів, а також забезпечити зручну навігацію та візуалізацію дизайну перед фактичною реалізацією вебсайту.

## РОЗДІЛ 3

### РЕЗУЛЬТАТИ ПРАКТИЧНОЇ РЕАЛІЗАЦІЇ ВЕБДОДАТКУ ДЛЯ СУПРОВОДУ НАВЧАННЯ ІНОЗЕМНІЙ МОВИ

#### 3.1 Функціональність інтерфейсу користувача вебдодатку

Після планування структури та вибору макету для створення майбутнього сайту. З використання інтегрального середовища програмування Visual Studio Code були розроблені всі складові проекту. Як було зазначено, вебдодаток реалізований у формі лендінгу, тобто навігація здійснюється зверху вниз поекранно між різними частинами додатку. При запуску відкривається верхня частина з трьома пунктами меню, банером і кнопкою переходу до наступної частини (див. рис. 2.5). Частина коду навігаційної панелі в завершеному вигляді показана на рис. 3.1.

```

<body class="
  <!-- HABI
  <header c
    <nav class="container mx-auto flex justify-between items-center p-4">
      <h1 class="text-2xl font-bold">LearnScots</h1>
      <ul class="flex space-x-6">
        <li><a href="#top" class="nav-link">Головна</a></li>
        <li><a href="#why-choose-us" class="nav-link">Про нас</a></li>
        <li><a href="#footer" class="nav-link">Наші контакти</a></li>
      </ul>
    </nav>
  </header>

```

A space-separated list of the classes of the element. Classes allows CSS and JavaScript to select and access elements via the [class selectors](#) or functions like the method `Document.getElementsByClassName()` .

[MDN Reference](#)

Рисунок 3.1 – Опрацювання панелі навігації

Окремо варто розібрати механізм навігації, оскільки саме в лендінгу користувачу має бути зручно пересуватися (рис. 3.2).

```

<h2 class="text-4xl font-bold mb-4">Опануйте шотландську мову за кілька тижнів</h2>
<p class="text-lg mb-6">Інтерактивні уроки, практичні вправи та підтримка викладачів.</p>
<button onclick="document.getElementById('plans').scrollIntoView({ behavior: 'smooth' })"
  class="btn-primary">Почати вже сьогодні</button>

```

Рисунок 3.2 – Код обробки події кнопки для навігації

Розбір за частинами коду (див. рис. 3.2) розкриває сутність застосованого набору елементів:

1. `<button>` - HTML-елемент кнопки. Це звичайна кнопка, яку можна натиснути.
2. `onclick="..."` - обробник події кліку. Це атрибут, який виконує JavaScript-код, коли користувач натискає на кнопку.
3. `document.getElementById('plans')` - цей виклик знаходить HTML-елемент на сторінці з `id="plans"`.
4. `scrollIntoView({ behavior: 'smooth' })` - це метод, який прокручує сторінку до вибраного елемента. `behavior: 'smooth'` - означає, що прокрутка буде плавною (анімованою), а не миттєвою. Це дуже корисно для створення scroll-ефекту на односторінкових сайтах, коли натискаєш на кнопку і тебе м'яко переносить до потрібного розділу.
5. `class="btn-primary"` - це клас для стилізації кнопки. Він визначений в CSS (рис. 3.3):

```
.btn-primary {  
  background-color: #3b82f6; /* синій */  
  color: white;  
  padding: 0.75rem 1.5rem;  
  border-radius: 9999px;  
  font-weight: bold;  
  transition: background-color 0.3s;  
}  
.btn-primary:hover {  
  background-color: #2563eb; /* темніший синій */  
}
```

Рисунок 3.3 – Опис стилю кнопки в частині CSS-файлу

Взагалі, стилі в даному проєкті були реалізовані через CSS-фреймворк Tailwind. Він стилізує елементи просто в HTML-коді за допомогою класів, що дозволяє не створювати окремі CSS-класи для кожного стилю. Завдяки цьому досягається певна швидкість розробки, що дозволяє також:

- не писати зайвого CSS;
- мати швидке прототипування: усе видно одразу в HTML;
- зміни відбуваються миттєво без перемикання між HTML і CSS.

Tailwind має вбудовану систему breakpoints (для мобільних, планшетів, десктопів), що дозволяє дуже просто керувати виглядом на різних екранах прямо в HTML.

Tailwind допомагає уникнути дублювання класів, завдяки чому CSS-файл мінімізується, і лишається тільки те, що використовується. Tailwind має дизайн-систему за замовчуванням: кольори, відступи, типографіка, шрифти вже узгоджені. Це гарантує єдність стилю без потреби щоразу вигадувати нове. Tailwind легко кастомізується через `tailwind.config.js`.

Tailwind має одну з найкращих документацій серед CSS-фреймворків і велику спільноту. Серед недоліків - HTML виглядає «захаращеним» (багато класів). Таким чином, можна вважати, що Tailwind – це CSS без проблем, ідеально для SPA, лендінгів, прототипів і навіть повноцінних проєктів.

Серед зручних елементів досягнутого UA-дизайну необхідно виділити слайдер та інформаційні блоки. Слайдер оформлений для перегляду відгуків про успішних слухачів курсу шотландської мови (рис. 3.4). Персони і фото в даному випадку взято зі штучних зображень. Текст придумано для прикладу.

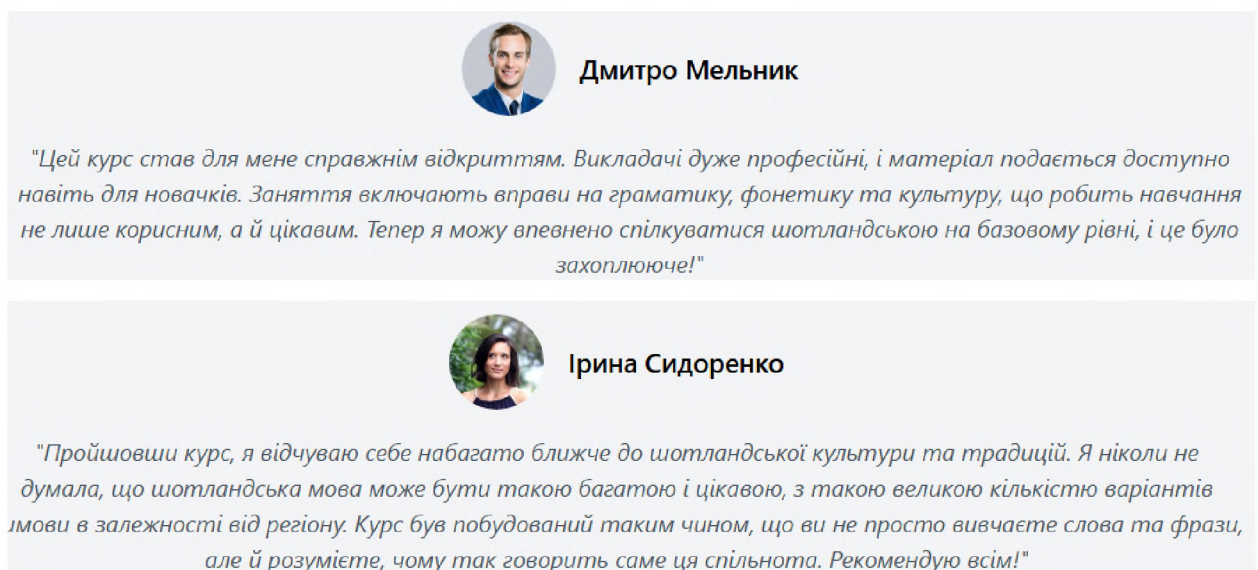


Рисунок 3.4 – Зображення двох фрагментів слайдера, які змінюються

Вся робота слайдера складається з  $i$  блоків (змінна  $index$ ) і підтримується за допомогою відповідного скрипта, записаного в JS-файлі (рис. 3.5).

```

1  const slider = document.getElementById("slider");
2  const slides = slider.querySelectorAll(".slide");
3  let index = 0;
4
5  function showSlide(i) {
6    const offset = -i * 100;
7    slider.style.transform = `translateX(${offset}%)`;
8  }
9
10 document.getElementById("prevBtn").addEventListener("click", () => {
11   index = (index - 1 + slides.length) % slides.length;
12   showSlide(index);
13 });
14
15 document.getElementById("nextBtn").addEventListener("click", () => {
16   index = (index + 1) % slides.length;
17   showSlide(index);
18 });
19
20 // Початкове відображення
21 showSlide(index);

```

Рисунок 3.5 – Повний JavaScript-код слайдера вебдодатку

Після слайдера розміщено інформаційний блок після заголовку другого рівня «Обирайте свій план навчання» про рівні курсів, використано спеціальні функції трансформації зображень (рис. 3.6).

### Обирайте свій план навчання




 <p><b>Basic – Ідеальний для знайомства</b></p> <p>Цей пакет підійде тим, хто хоче спробувати основи шотландської мови без значних</p>	 <p><b>Pro – Для серйозного навчання</b></p> <p>Оптимальний вибір для тих, хто прагне впевнено спілкуватися шотландською. Цей пакет</p>	 <p><b>Ultimate – Повне занурення</b></p> <p>Для тих, хто хоче опанувати мову на високому рівні та відчувати культуру Шотландії.</p>
---	---	---

Рисунок 3.6 – Збільшення зображення (висування) при наведенні мишею

Нижня інформаційна частина вебдодатку зайнята блоком з медіаелементом – відео з посиланням на YouTube (рис. 3.7).



Рисунок 3.7 – Розміщення на сайті посилання на відеоресурс - сучасно

Отже, після опису інтерфейсу користувача можна виділити, що сайт зроблений добре, додані до нього анімації, розписана тільки головна інформація без зайвих слів. Кожен з динамічних елементів вебдодатку – слайдер, інформація, відео розміщені в блоках, так само як і статті, секції. Все це полегшує не лише сприйняття всього додатку як цілісного продукту, має єдиний стиль, але й полегшує управління всією сторінкою. Зокрема дозволяє проводити заходи оптимізації, управління контентом і змінами.

### 3.2 Окремі заходи оптимізації коду і функцій вебдодатку

Після написання остаточної версії коду та перевірки вебдодатку в браузері виявлені досить гарний лаконічний дизайн, досягнуто функціональність. Однак, кожного разу виникають ідеї для покращення. Зокрема, прискіпливо необхідно доопрацювати елементи, які впливатимуть на пошукову оптимізацію та адаптивність.

Наприклад, вирішено додати meta description, Open Graph теги (og:title, og:image, og:description) – це важливо для пошуковиків та соціальних мереж.

Онією з умов успішного просування контенту вебдодатку є його адаптивність, тобто можливість переглядати і використовувати функції через мобільні пристрої. Кілька елементів забезпечують адаптивність. Наприклад, тег `<meta name="viewport" content="width=device-width, initial-scale=1.0" />`.

Також в самому HTML в частині `<body>` записано спеціальний клас: `<body class="bg-sky-100 text-gray-900 flex flex-col items-center p-4">`.

Приклад роботи на екрані смартфона показано на рис. 3.8.

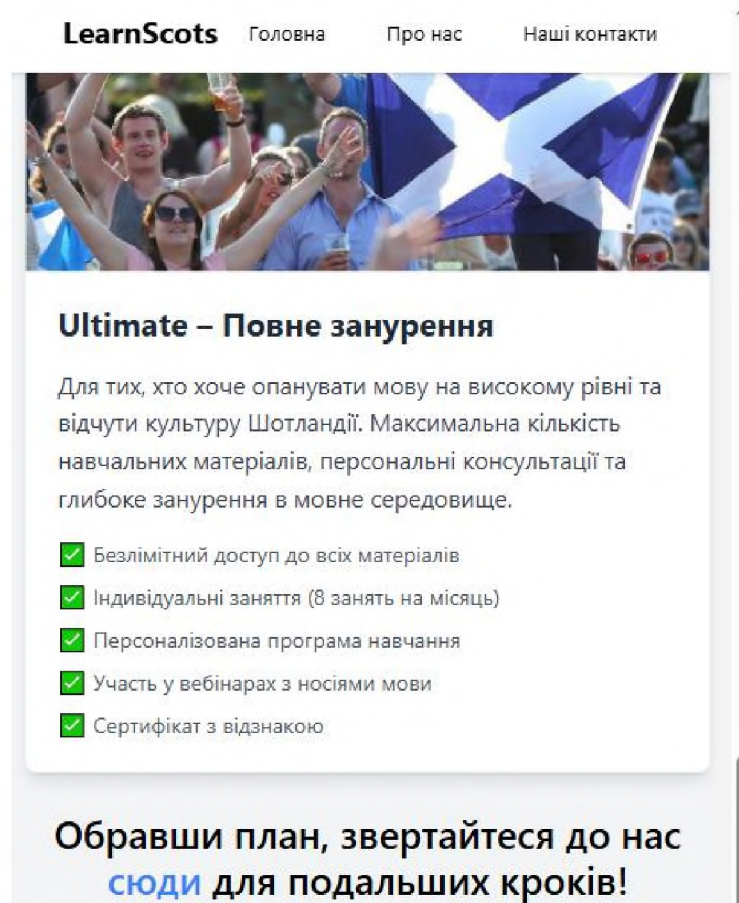


Рисунок 3.8 – Перевірка адаптивності вебдодатку на екрані смартфона

Як видно, всі блоки вишикувалися в одну колонку згідно вертикальної розмітки згори-вниз». Функціональність та властивості кожного блоку

збереглися. Елементи футера залишилися всередині цього блоку і також змінили порядок розташування на вертикальний (рис. 3.9).

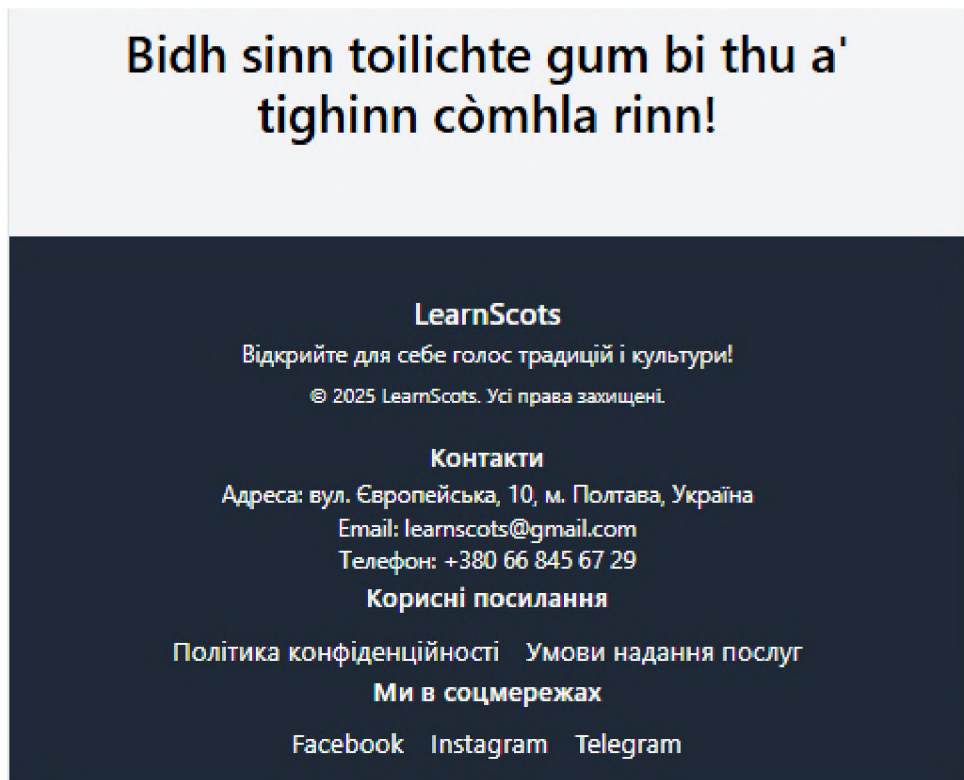


Рисунок 3.9 – Вигляд футера при стисненні на вертикальному екрані

Таким чином, адаптивність всього інтерфейсу забезпечена. Серед інших поправок – переключення мови з «ua» на «uk» в тезі `<html lang="ua">`, що також сприятиме кращому пошуку. В майбутньому заплановано додати кілька мовних інтерфейсів із використанням фреймворку Alpine.js, а також інші.

### 3.3 Економічна оцінка вартості розроблення вебдодатку

Модель оцінки вартості програмного забезпечення є набором функцій, які визначають зв'язок між характеристиками проекту та витратами, необхідними для його реалізації. Існують різні типи таких моделей, включаючи лінійні, мультиплікативні, степеневі, емпіричні та аналітичні моделі. Дві найбільш поширені та добре задокументовані алгоритмічні моделі

- це модель Путнема (степенева, аналітична) і модель COCOMO (степенева, емпірична) [41]. Аналізуючи ці моделі оцінки відносно розроблюваного вебсайту, було вирішено вивести спрощену формулу. Тому, для розрахунку економічної оцінки вартості розробки вебдодатку буде використана наступна формула:

$$E_e = (T \times C) + B_I + B_M + B_O , \quad (3.1)$$

де  $E_e$  – економічна оцінка вартості;

$T$  – трудові витрати – це кількість годин, витрачених на розробку вебдодатку;

$C$  – ставка роботи (середня ставка оплати праці для фахівця);

$B_I$  – витрати на інфраструктуру це витрати на необхідне обладнання, програмне забезпечення та інфраструктуру для розробки вебпрограми;

$B_M$  – витрати на маркетинг це витрати на рекламу та просування вебдодатку;

$B_O$  – витрати на обслуговування це витрати на підтримку, оновлення та обслуговування вебдодатку після його розробки.

Для проведення оцінки вартості роботи в грошовому еквіваленті необхідно використати дані з популярних довідкових інтернет-джерел про рівень оплати праці вебдизайнера, графічного дизайнера та інших фахівців [42]. Щоб зорієнтуватися в шкалі оплати, необхідно також оцінити кваліфікаційний рівень виконаної роботи. В описаному проєкті були застосовані навички верстки за допомогою таких вебтехнологій, як HTML&CSS, програмування на JavaScript, створення графічного прототипування макету в Figma. Отримано якісний UI/UX дизайн, інтерактивні елементи, проведено технічну оптимізацію вебдодатку. Така робота може бути прирівняна до досвідченого рівня Junior, заробітна плата якого становитиме біля \$800 на місяць, що в еквіваленті складає близько 33120 грн/міс, або 190 грн/год (курс НБУ \$1 = 41,4 грн станом на 25.05.2025).

Для розробки даного проєкту знадобиться не менше 15 робочих днів, або 120 години. Для розрахунків за формулою (3.1) дані були внесені до табл. 3.1.

Таблиця 3.1. – Дані для розрахунку вартості розробки вебдодатку

№	Назва ресурсу, послуги, змін роботи	Кількісний показник витрат
1	Трудові витрати	120,00 год.
2	Ставка роботи	190,00 грн/год.
3	Витрати на інфраструктуру (інтернет)	350,00 грн
4	Витрати на маркетинг	1300,00 грн
5	Витрати на обслуговування	1000,00 грн

Ці дані основані на середніх значеннях оцінки роботи веброзробників на основі довідкових сайтів, наприклад [42]. Отже, після отримання даних можна провести розрахунки:

$$E_e = (120 \times 190) + 350 + 1300 + 1000 = 25450$$

Отже, розрахунки показали загальну оцінку вартості розробки застосунку. Дана вартість вийшла з урахуванням всіх основних факторів, які могли би вплинути на неї. Так як ця робота була розроблена для прикладу, вона вийшла більш дешевою. За припущенням дизайнер працював на фрілансі, тому не була врахована оренда приміщення, комунальні витрати. Розробка аналогічного вебдодатку при замовленні в компанії може коштувати більше 30000 грн.

## ВИСНОВКИ

У результаті виконання роботи була досягнута її початкова мета та вирішені поставлені завдання. Було проаналізовано основну наукову, методичну та нормативну літературу з теми розробки та проєктування навчальних вебдодатків.

У першому розділі виконано визначення основних понять з вебпрограмування, проведено визначення основних технологій з веброзробки, виділено історію їх впровадження в сучасні інформаційні технології. Проведено аналіз видів основних типів вебдодатків інформаційно-довідкового характеру, акцентовано увагу на особливості кожного з них. Виконано аналіз існуючих вебсайтів інформаційно-довідкового характеру. Було висвітлено особливості різних сучасних технологій веброзробки і взаємозв'язок. Виділено основні переваги та недоліки.

У другому розділі було визначено характеристики запланованого для реалізації вебдодатку для супроводу вивчення іноземної мови, щоб відобразити його в найбільш кращому вигляді. Було обґрунтовано вибір мов розмітки HTML та CSS, вебпрограмування на JavaScript, обґрунтовано застосування інших засобів, наприклад, фреймворків.

Було проведено доцільне планування структури по розміщенню матеріальної бази на вебсайті та планування і розробку найбільш доречного макету, виходячи з раніше згаданого планування структури довідника. Було розглянуто створення вебсайту на базі додатку Visual Studio Code.

У третьому розділі було висвітлено вже готову розробку у вигляді інтерактивного вебдодатку для зручного користування при вивченні іноземної мови на прикладі шотландської мови. Детально представлено методи кодування інтерфейсу користувача. Був висвітлений основний функціонал сайту, навігацію та його особливості в виді анімацій та коду до них.

За підсумками виконаної роботи проведений економічний аналіз розробки подібних вебдодатків та розрахована оцінка вартості проєкту, в яку

входять трудові витрати, ставка роботи, витрати на інфраструктуру, витрати на маркетинг та витрати на обслуговування що й були представлені у вигляді таблиці. Розрахунок проводився як для рівня вебдизайнера Junior з гарними навичками. Оціночна вартість роботи підтверджує широкі можливості для роботи професіонала в області вебдизайну, можливості отримати гідну працю.

Таким чином, поставлені задачі кваліфікаційної роботи розв'язано у повному обсязі, мета досягнута.

Розробка може бути використана школою або студією іноземних мов для ознайомлення зі змістом курсу, запису на пробні уроки або співбесіди, відгуками користувачів, історією мови, яка вивчається. Такий додаток може бути цікавий для української аудиторії, а також доопрацьований з інтерфейсом на інших мовах. Подальший розвиток проєкту може полягати в додаванні ефективної бекенд-частини, розширення пропозицій та інформації про тарифи, інтеграцією з платіжними системами тощо.