

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
Кафедра _____ Медіасистем та технологій _____
Рівень вищої освіти _____ перший (бакалаврський) _____
Спеціальність _____ 186 Видавництво та поліграфія _____
Тип програми _____ Освітньо-професійна _____
Освітня програма _____ Видавничо-поліграфічна справа _____
(шифр і назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри МСТ _____
(підпис)
« 22 » травня 2023 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

студентові _____ Батраченко Денис Анатолійович _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Проектування та розробка web-сайту «Національна кухня України» _____

Затверджена наказом по університету від _____ 22 травня 2023р. № 506 Ст _____


2. Термін подання студентом роботи до екзаменаційної комісії _____ 15 червня 2023 р. _____

3. Вихідні дані до роботи
Національні та міжнародні стандарти оцінки якості веб-сайтів; ДСТУ Інженерія систем і програмних засобів. Розроблення та керування WEB-сайтами для систем, програмних засобів та інформаційних послуг

4. Перелік питань, що потрібно опрацювати в роботі
Вступ; Аналіз технічного завдання; Аналітичний огляд літератури за темою; Аналіз аналогів; Опис практичної частини; Засоби розробки веб-сайту; Економічна частина; Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п. 5 включається до завдання за рішенням випускової кафедри)
Тема роботи; Актуальність теми; Мета роботи; Завдання кваліфікаційної роботи; Програмне забезпечення; Цільова аудиторія; Аналіз завдання і огляд літератури за темою; Аналіз аналогів; Практична частина; Домашня сторінка; Сторінка міста; Сторінка усі Міста; Сторінка про автора; Економічна частина; Висновки.

6. Консультанти розділів роботи (п. 6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п. 1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Основна частина	проф. Манаков В. П.		06.06.2023
Економічна частина	ас. Помогалова Н.В.		25.05.2023

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз технічного завдання	18.05.2023	виконано
2	Аналітичний огляд літератури за темою	20.05.2023	виконано
3	Аналіз аналогів	22.05.2023	виконано
4	Технології розробки	24.05.2023	виконано
5	розробка практичної частини	26.05.2023	виконано
6	Економічна частина	28.05.2023	виконано
7	Оформлення пояснювальної записки	01.06.2023	виконано
8	Оформлення графічної частини	01.06.2023	виконано

Дата видачі завдання 22 травня 2023 р.

Студент 
(підпис)

Батраченко Д.А.

Керівник роботи 
(підпис)

проф. Манаков В.П.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 59 с., 10 табл., 13 рис., 1 дод., 16 джерел.

WEB-САЙТ, КУХНЯ УКРАЇНИ, ОПТИМІЗАЦІЯ, РЕЦЕПТИ, HTML, CSS, JAVASCRIPT, JQUERY, ПРОЕКТ, ПРОГРАМА, АДАПТИВНІСТЬ, VISUAL STUDIO CODE, SUBLIME TEXT, ВЕРСКА, ДИЗАЙН, НАВІГАЦІЯ.

Мета роботи полягає в проектуванні дизайну на розробці web-сайту «Національна кухня України», який буде надавати інформацію про популярні рецепти та методи приготування культових страв України.

Проведено аналіз технічного завдання. Кваліфікаційна робота складається з декількох частин. На початку визначається мета завдання та предметна область. Потім проводиться аналіз і вибір програмного забезпечення, для вирішення завдання з розробки сайту. Останній етап включає проектування та розробку клієнтської частини сайту.

Проаналізована література за темою і виявлено, що незалежно від розміру майбутнього проекту, важливо ретельно продумати кожен етап створення веб-сторінки та дотримуватися графіка, щоб продукт був успішним. Проаналізовані сайти конкурентів. На стадії вибору засобів розробки, обрано програми, а також основні технології у верстці. Далі були створені дизайн та макет сайту.

Перед початком проектування та розробки виконується оцінка економічної ефективності сайту. Це дозволяє прогнозувати потенційні результати, та обґрунтувати доцільність розміщення сайту в Інтернеті. Собівартість розробки та ціна є основними економічними показниками, які враховуються при розрахунку потенційного прибутку.

ABSTRACT

Explanatory note of the qualification work: 59 p., 10 tabl., 13 pic., 1 app., 16 sources.

WEB-SITE, UKRAINIAN CUISINE, OPTIMIZATION, RECIPES, HTML, CSS, JAVASCRIPT, JQUERY, PROJECT, PROGRAM, ADAPTERS-INPUT, VISUAL STUDIO CODE, SUBLIME TEXT, EDITING, DESIGN, NAVIGATION.

The purpose of the work is to design a design for the development of the "National Cuisine of Ukraine" website, which will provide information on popular recipes and methods of cooking cult dishes of Ukraine.

Analysis of the technical task was carried out. The qualification work consists of several parts. At the beginning, the goal of the task and the subject area are determined. Then the analysis and selection of software is carried out to solve the task of developing the site. The last stage includes design and development of the client part of the site.

The literature on the topic was analyzed and it was found that regardless of the size of the future project, it is important to carefully think through each stage of creating a web page and adhere to the schedule in order for the product to be successful. Competitors' sites were analyzed. At the stage of selection of development tools, programs are selected, as well as the main technologies in the layout. Next, the design and layout of the site were created.

Before starting design and development, an assessment of the site's economic efficiency is performed. This allows you to predict potential results and justify the expediency of placing the site on the Internet. Development cost and price are the main economic indicators that are taken into account when calculating potential profit.

ЗМІСТ

	С.
ВСТУП	7
1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ.....	9
2 АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРИ ЗА ТЕМОЮ	11
2.1 Поняття веб-сайт та веб-додаток.....	11
2.2 Редактори коду	12
2.3 Основні етапи створення сайтів	14
2.4 Основні вимоги до сучасних веб-сайтів.....	16
3 АНАЛІЗ АНАЛОГІВ.....	21
4 ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ.....	26
4.1 Розробка HTML структури.....	27
4.2 Розробка CSS стилізація	29
4.3 Препроцесор CSS	31
4.4 Мова програмування JavaScript.....	32
4.5 Бібліотека jQuery	33
4.6 Система управління версіями Git.....	35
5 РОЗРОБКА ПРАКТИЧНОЇ ЧАСТИНИ.....	37
5.1 Реалізація задач та мети	37
5.2 Інструментальні засоби розробки сайту.....	37
5.3 Проектування структури сайту.....	39
5.4 Створення графічного дизайну.....	41
5.5 Розміщення інформації у блоках.....	45
6 ЕКОНОМІЧНА ЧАСТИНА	47
ВИСНОВКИ.....	57
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	58
ДОДАТОК А Дизайн сайту.....	60

ВСТУП

В наш час розробка веб-додатків є однією з перспективних сфер діяльності для багатьох компаній, що спеціалізуються на високотехнологічних цифрових та комп'ютерних технологіях. Веб-додаток є прикладним програмним забезпеченням, яке працює на основі розподіленої логіки між сервером і клієнтом, а передача інформації між ними здійснюється по мережі. Клієнтська частина програми відповідає за інтерфейс користувача, а серверна частина отримує і обробляє запити від клієнта, виконує обчислення, формує веб-сторінки і відправляє їх клієнту за допомогою протоколу HTTP.

Інтернет став основним джерелом інформації, замінюючи традиційні форми зв'язку, такі як телебачення, радіо та друковані видання. Роль Інтернету у цьому контексті постійно зростає.

Для компаній, які прагнуть покращити свій маркетинг, збільшити продажі та забезпечити високу якість послуг, використання Інтернету стає необхідністю. Одним із важливих завдань є поширення інформації серед потенційних споживачів, яке має здійснюватись через Інтернет. Корпоративні портали надають можливість доставки цієї інформації кінцевим користувачам. На таких порталах користувачі можуть ознайомитися з останніми новинами компанії, дізнатися про її продукцію та розробки.

Для створення веб-програми необхідно вирішити наступні завдання:

- вивчити доступні інструменти та інформаційні технології для розробки Веб-додатків;
- визначити структуру та дизайн веб-додатку, що розробляється;
- розробити основний функціонал клієнтської частини веб-додатку;
- забезпечити адаптацію веб-програми для роботи на різних пристроях;
- провести функціональне тестування веб-програми.

Метою кваліфікаційної роботи є розробка веб-сайту «Національна кухня України». Цей сайт створений для того, щоб надавати велику кількість інформацію про національну кухню України, включаючи її традиції, особливості регіональних кулінарних стилів, популярні інгредієнти та спеції, а також сезонні страви. Відвідувачі сайту зможуть дізнатися про типові методи приготування та популярні рецепти. Сайт буде наслідувати всі сучасні тенденції веб-розробки, мати приємний дизайн, інтуїтивно зрозумілу навігацію, також сайт буде кросбраузерним, тобто буде відображатися і функціонувати ідентично у всіх браузерах, що часто використовуються. А також сайт буде адаптивним, тобто буде повинен однаково добре відображатися на екрані комп'ютера, планшета, смартфона.

Кваліфікаційна робота складається із шістьох розділів. Перша розділ включає постановку мети, завдань і опис предметної області. Другий розділ складається з аналітичного огляду літератури за темою. Третій розділ включає аналіз аналогів. Четвертий розділ складається з аналізу аналогів. Четвертий розділ описує вибір технологій розробки. П'ятий розділ складається з опису реалізації поставлених цілей. Шостий розділ це оцінка економічної ефективності, яка прогнозувати потенційні собівартості розробки сайту.

1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

Створення сайту про національну кухню української включає кілька кроків. Починати потрібно з визначення мети майбутньої роботи, наприклад, надання інформації про рецепти, історію, рекомендації ресторанів. Після цього потрібно визначити цільову аудиторію сайту, визначте, кому він буде призначений, туристам, любителям готувати, українцям чи людям, які живуть за межами України. На наступному етапі потрібно вибрати назву сайту, вибрати слід цікаве ім'я, яке буде відображати його тему і бути незабутнім. Далі потрібно продумати структуру, розробити дизайн цього сайту, який відповідатиме його тематиці. Розмістити інформацію про рецепти, історію, фотографії та будь-яку іншу релевантну інформацію. Після цього необхідно розробити функціональність: визначте, яку функціональність хотілося б бачити на веб-сайті. Наприклад це можуть бути фільтри на кшталт страви, пошук за рецептами, можливість додавати коментарі та відгуки, створення профілів користувачів та інше. Після цього слід перейти до створення унікального та інформативного контенту про українську національну кухню. Це можуть бути статті про історію кухні, рецепти з покроковими інструкціями, фотографії та відео.

Цільова аудиторія сайту про українську національну кухню є дорослі люди віком від 25 до 54 років. Це широка вікова група, яка зазвичай цікавиться гастрономією, культурою та подорожами. Вони можуть шукати рецепти, історичну інформацію, рекомендації ресторанів та іншу інформацію про українську кухню. Сайт спеціально був зроблений англійською мовою, щоб люди різних культур та національностей могли б познайомитися зі стравами та їх рецептами. Люди, які активно подорожують або цікавляться культурою та кулінарією, можуть бути зацікавлені у вивченні української національної кухні. Вони можуть шукати інформацію про регіональні страви, типові інгредієнти та унікальні кулінарні традиції.

Переходячи до практичної частини, щоб створити сайт про українську національну кухню, потрібно придумати і розробити макет, створити естетично привабливий дизайн сайту, що відповідає його тематиці. Розмістити інформацію про рецепти, історію, фотографії та інший контент, який буде цікавий цільовій аудиторії.

Результатом роботи, має стати розроблений веб-сайт який буде відповідати всім сучасним вимогам у сфері зручність користування, а також буде добре виглядати, з правильно підібраними поєднаннями кольорів, шрифтів, іконок і кнопок.

2 АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРИ ЗА ТЕМОЮ

2.1 Поняття веб-сайт та веб-додаток

Все, що ми відкриваємо у браузері, часто називають сайтами, проте це не зовсім точне визначення. Сайт можна розглядати як місце, де користувачі отримують інформацію, читають, слухають та дивляться її. Однак, крім сайтів, існують веб-додатки, які дозволяють користувачам взаємодіяти з інтерфейсом. Наприклад, веб-програми дозволяють робити оплату, оформляти бронювання або грати в ігри. Вся інформація, яка з'являється у веб-застосунку, залежить від запиту користувача. Ось кілька прикладів веб-додатків: ігри, інтернет-магазини, соціальні мережі, освітні продукти, редактори фото, відео та тексту, системи бронювання. Вони мають складнішу структуру порівняно із звичайними інформаційними сайтами. Замість пасивного читання, користувач стає активним учасником бізнес-процесу, взаємодіючи з компанією. Хоча інформаційні сайти без інтерактивності все одно потрібні для бізнесу, їх можливості обмежені. Наприклад, це може бути просто сайт-візитівка.

Існує безліч способів розробки додатків, і вибір конкретного типу програми впливатиме на його вартість, терміни розробки та функціональність. Давайте розглянемо кожний тип і визначимо, для яких завдань буде найбільш підходяща певна архітектура.

Насамперед, програми можна розділити на дві категорії:

- розроблювані з використанням коду;
- створювані за допомогою конструкторів.

Сучасні редактори, такі як Webflow, надають можливість створювати інтерактивні рішення, наприклад, підключати платіжні системи та створювати функціональні інтернет-магазини. Цей варіант обирають тому, що це швидко та економічно вигідно.

Веб-додатки, написані кодом, розробляються в редакторах коду за допомогою HTML, CSS та JavaScript.

2.2 Редактори коду

Visual Studio Code – це універсальний редактор коду, розроблений компанією Microsoft. Він має високу швидкість роботи і великий набір функцій. Це середовище з відкритим вихідним кодом, яке підтримує автоматичне інтелектуальне доповнення коду за допомогою технології Microsoft IntelliSense, а також підсвічування синтаксису для більшості популярних мов програмування. Visual Studio Code має достатній набір можливостей як для аматорської, так і професійної розробки. Вбудовані інструменти дозволяють працювати з Git репозиторіями та виконувати рефакторинг коду. Крім того, функціональність редактора можна розширити за допомогою тисяч плагінів.

Переваги Visual Studio Code:

- Visual Studio Code – це редактор, який працює на Windows, Linux і MacOS, що дозволяє вам працювати незалежно від платформи, на якій ви працюєте;
- при відвідуванні веб-сайту Visual Studio Code ви миттєво помітите, що він підтримує практично всі основні мови програмування. Він включає підтримку Python, JavaScript, HTML, CSS, TypeScript, C++, Java, PHP, Go, C#, SQL, Ruby, Objective-C та багатьох інших;
- VSCode постачається з вбудованим засобом налагодження, який є однією з його ключових функцій. Це допомагає прискорити цикл редагування, компіляції та налагодження для будь-якого програміста. За замовчуванням він поставляється з налагоджувачем, який підтримує Node.js, що дозволяє налагоджувати будь-який код, що виконується на JavaScript. Однак, ви також можете використовувати розширення для інших середовищ виконання.

Sublime – це легкий, але потужний редактор, що ідеально підходить як для новачків у розробці, так і для досвідчених професіоналів. Він швидко запускається навіть на слабких комп'ютерах та підтримує плагіни, написані мовою Python. Sublime пропонує гнучкі налаштування, що дозволяють зручно працювати з проектами, що складаються з тисяч файлів. Редактор забезпечує автодоповнення, можливість множинного редагування та миттєвої зміни кількох рядків коду. Всі ці функції допомагають скоротити рутинні завдання та мінімізувати можливість виникнення нових помилок.

Переваги Sublime:

- редактор надає інтерфейс, що дозволяє налаштовувати його зовнішній вигляд під конкретні завдання. Наприклад, для роботи з кількома файлами одночасно можна налаштувати режим, який показує до чотирьох панелей одночасно. Також доступний повноцінний режим, який відображає одну робочу область без додаткових вікон та панелей;

- функція множинного виділення дозволяє одночасно редагувати кілька фрагментів коду, використовуючи покажчики, що у різних частинах коду. Можливе і одиночне редагування з швидким перемиканням на множину. Це дозволяє швидко редагувати складний структурований код без використання макрокоманд і прискорює процес написання коду;

- редактор автоматично пропонує варіанти закінчення запису залежно від вибраної мови програмування. Крім того, програма автоматично завершує створені програмістом змінні. Це прискорює написання коду та знижує ймовірність помилок.. Однак, ви також можете використовувати розширення для інших середовищ виконання;

- для зручності роботи з фрагментами коду, що повторюються, розробники можуть зберігати їх в Sublime Text і повторно використовувати. Можна також зберігати ключові слова для швидкого запуску цих фрагментів.

2.3 Основні етапи створення сайтів

Розробка веб-сайту вимагає виконання значного обсягу робіт, незалежно від розміру майбутнього проекту. Тому, для успішної реалізації продукту, важливо детально продумати всі етапи створення веб-сторінки та дотримуватись запланованого плану. Етапи розробки сайту:

- етап визначення тематики та основної мети проекту;
- етап розробка технічного завдання;
- етап прототипування, макетування та дизайн;
- етап верстка та програмування;
- етап наповнення контентом;
- етап тестування;
- етап здача готового проекту.

Спершу слід чітко визначити, для яких цілей потрібен сайт. Від цього залежатимуть портрет цільової аудиторії, тип ресурсу та основні вимоги. Розуміння мети та остаточного результату допоможе побудувати структуру проекту і визначити необхідні етапи створення сайту для досягнення цілей. Усі ці особливості слід обговорювати на початковому етапі. Активна участь у визначенні мети допоможе розробити концепцію, сформулювати основні цілі та обрати інструменти для їх досягнення. Тільки після визначення основних пріоритетів можна переходити до наступного етапу.

Технічне завдання є офіційним документом, який служить основою для подальшої роботи. У ньому визначаються всі деталі: структура сайту (кількість сторінок, розділів, категорій, блоків), вимоги до дизайну, функціоналу, візуального та текстового наповнення, а також технічні можливості. План розробки сайту або ТЗ повинен відповідати наступним вимогам:

- детальність. Усі аспекти та кроки, які будуть виконуватися фахівцем, докладно прописуються;
- чіткість. У цьому документі не місце для суб'єктивних формулювань, вимоги повинні бути однозначними і зрозумілими;

– зрозумілість. Всі вимоги розписуються зрозумілою для фахівця мовою, використовуючи відповідну термінологію.

Технічне завдання є інструкцією, яка буде постійно використовуватися під час розробки сайту. Перехід до основних робіт можливий лише після узгодження всіх питань.

На етапі, який включає прототипування, макетування та дизайн, створюється макет, що перетворює ідеї в реальний об'єкт. Мета – не створення повноцінного готового до роботи веб-інструмента, але ви зможете ознайомитися з ним і оцінити його переваги. На цьому етапі команда працює над кількома ескізними варіантами, виходячи з технічного завдання. Багато людей помилково вважають, що цей етап стосується лише зовнішнього оформлення сторінок. Насправді спеціалісти надають найбільшу увагу правильному розташуванню елементів з урахуванням правил юзабіліті та інших технічних особливостей. Завдяки врахуванню всіх цих етапів створення веб-сайту можна забезпечити швидку та стабільну роботу ресурсу. Після підготовки прототипів та макетів, треба узгодити їх. Якщо потрібно, вносяться необхідні зміни, поки проект не буде остаточно затверджений.

Наступним кроком є технічна складова. На цьому етапі дизайн поєднується з двигуном, перетворюючи сайт у робочий інструмент з функціональністю. Команда спеціалістів використовує знання основ HTML, підключає CSS стилі та інтегрує з CMS. Варто зазначити, що не всі сайти створюються на основі CMS. Деякі прості односторінкові ресурси можуть функціонувати без системи управління контентом. Далі в процесі створення веб-сайту надаються послуги з програмування. Фахівець "оживляє" сайт та додає необхідну функціональність. У більшості випадків програмування здійснюється на основі CMS, таких як WordPress, або сервіс Тільда, але іноді потрібно писати код з нуля для створення унікальних функцій або блоків.

Після завершення верстки, ми отримуємо фактично робочий інструмент, але з порожніми розділами та сторінками. Ці порожні розділи та сторінки потрібно заповнити текстовим та графічним контентом. Важливо,

щоб контент відповідав стандартам оптимізації для подальшого просування ресурсу в пошукових системах.

Узагалі, ми уже розглянули основні етапи створення сайту, які стосуються безпосередньо процесу розробки. Тестування є останнім етапом, включаючи проведення різних видів перевірок для виявлення помилок, некоректного функціонування та загальної працездатності ресурсу. Виявлені помилки виправляються фахівцями до повного усунення. Ще один важливий аспект – розміщення сайту в Інтернеті. Коли ми отримуємо готовий проект, який готовий до роботи, він негайно не почне приносити прибуток. Для цього сайт має бути доступним для користувачів, тому його переносять на постійне місце розміщення – хостинг. Крім того, потрібно вибрати доменне ім'я для сайту. Підключення домену до хостингу – крок, який можна виконати в будь-який момент. Зазвичай це питання вирішується на початковому етапі. Після розміщення сайту в Інтернеті проводиться фінальне тестування, щоб перевірити його працездатність.

Після передачі готового проекту замовнику, фахівці надають навчання щодо роботи з сайтом, що допомагає клієнту самостійно оновлювати інформацію на ресурсі, збирати аналітику, вносити необхідні зміни та приймати рішення. Залежно від побажань клієнта, можна продовжувати співпрацю з розробником, оскільки будь-який веб-ресурс вимагає подальшого розвитку, підтримки та просування. Варто відзначити, що багато залежить від типу сайту. Наприклад, сайт-візитка після завершення розробки та розміщення на хостингу не потребує особливої уваги або регулярного оновлення інформації. З іншого боку, інтернет-магазин потребує постійного оновлення та підтримки для збереження своїх позицій у пошукових системах.

2.4 Основні вимоги до сучасних веб-сайтів

Розвиток веб-розробки відбувається з величезною швидкістю, порівняно з іншими науково-технічними областями. За кілька років прості

статичні веб-сторінки, створені з використанням звичайного HTML, перетворилися на складні, багатофункціональні веб-системи, інтегровані з іншими додатками. Компанії тепер мають можливість перенести в інтернет безліч бізнес-процесів. Сучасні компанії створюють не один, а безліч веб-сайтів, кожен із яких вирішує свої унікальні завдання.

Останнім час все більше компаній усвідомлюють, що сайт – це не просто електронна візитка чи онлайн-каталог, а зручний та дуже ефективний інструмент бізнесу. Однак, щоб сайт став дійсно ефективним, необхідно якісно розробити його та просувати правильно. Оцінка якості сайту не така проста задача, як може здатися на перший погляд. Більшість людей взагалі не уявляють, що можна оцінювати, окрім дизайну. Хтось, звичайно, згадує зручну структуру сайту, але як визначити, чи вона зручна для всіх користувачів. Інші важливі моменти рідко згадуються. Якісний сайт повинен не тільки відповідати всім сучасним технічним вимогам та очікуванням цільової аудиторії, а й враховувати внутрішні та зовнішні фактори ранжування (критерії якості сайту з погляду пошукових систем). Для того, щоб створити якісний, сучасний та ефективний сайт необхідно:

- необхідно створити дизайн, який не тільки ефективний та привабливий, але й відповідає очікуванням цільової аудиторії та відображає фірмовий стиль компанії. Якщо вигляд сайту виглядає добре, відвідувачі будуть вважати, що компанія також добре функціонує;

- необхідно ретельно продумати "юзабіліті" сайту, тобто його структуру та зручність використання. Лише привабливого дизайну недостатньо. Кожна сторінка сайту повинна максимально відповідати очікуванням відвідувача, всі елементи сторінки повинні бути розташовані у зручних місцях, інформація на сайті має бути легко читаною, а навігація та пошук – інтуїтивно зрозумілими та простими. Відвідувач повинен одержувати те, що шукав на сайті за мінімальний час. В ідеалі він повинен негайно зробити замовлення або виконати іншу необхідну цільову дію;

– налаштувати автоматичну адаптивність сайту для різних пристроїв, таких як телефони, ноутбуки, планшети та інші. Адаптивна верстка або дизайн є створення сторінок сайту, які автоматично пристосовуються під розмір, роздільну здатність та орієнтацію екрану конкретного пристрою. Для кожного пристрою готуються оптимізовані дизайни, які автоматично завантажуються під час відвідин сайту. Головна мета полягає в тому, щоб забезпечити зручність використання сайту як на комп'ютерах, так і смартфонах та планшетах. Для цього дизайнер створює кілька макетів майбутнього сайту для різних типів та моделей мобільних та стаціонарних пристроїв. Наприклад, сторінка зі складним дизайном може мати шість або більше варіантів макета, у той час як у простих дизайнів може бути лише два макети – вертикальний та горизонтальний. Розробник під час верстки задає відповідні розміри та роздільну здатність екрана для кожного макета. Коли відвідувач заходить на сайт, система визначає, з якого пристрою відкрито сторінку та відображає відповідний макет інтерфейсу;

– основна складова успішного просування сайту – це розміщення виняткового та високоякісного контенту на ньому. Контент є ключовим фактором у приверненні уваги користувачів. Тільки сайт, що пропонує цікавий, корисний і унікальний контент, має можливість не тільки потрапити в верхню частину пошукових результатів, але й утриматися на високих позиціях протягом тривалого часу;

– розробляти високоякісні сторінки для кожного пошукового запиту є важливою практикою. Багато хто вже усвідомив, що направляти всіх відвідувачів на головну сторінку сайту неефективно, але мало хто замислюється над необхідністю створення окремих сторінок, або сторінок приземлення для кожного контекстного оголошення або фрази, за якими просувається сайт у пошуковій видачі. Ці сторінки повинні швидко та повноцінно відповідати на запити користувачів, за якими вони переходять на сайт. Після відвідування такої сторінки у відвідувача не повинно виникнути бажання повернутися до пошукової видачі та переглянути інші сайти,

включаючи конкурентів. Тому важливо чітко уявляти, що саме очікує побачити відвідувач при кожному ключовому запиті та створювати окремі "спеціальні" сторінки, що відповідають цим запитам;

– сайт повинен мати можливість залишати відгуки та інтегруватися з соціальними мережами. На сайті має бути функція, яка дозволяє клієнтам ставити вподобання та залишати відгуки про компанію та її товари. Оскільки пошукові системи часто виключають із індексу однотипні сторінки, які здаються неінформативними, відгуки на веб-сайтах підвищують унікальність сторінки. Присутність посилань на соціальні мережі, кількість вподобань, та наявність відгуків, дозволяють пошуковій системі оцінити соціальну активність компанії та у відповідь "підняти" позиції сайту в результатах пошуку;

– забезпечити кросбраузерність, тобто властивість веб-ресурсу, що забезпечує його однакове і безперебійне відображення у всіх популярних браузерах, без помилок у верстці та з правильним відображенням контенту. Цей аспект є важливим для пошукових систем, так і для користувачів. Відсутність проблем із версткою означає, що банери, зображення, логотипи не перекривають одне одного і коректно відображаються. Текстова інформація повинна бути легко читається, особливо у таблицях. Також важливо, щоб веб-додаток правильно функціонував із сервісними програмами, вбудованими на сайті;

– гарантувати захист від спаму, швидке завантаження сторінок сайту, захист від взломів та вірусів є важливим аспектом. Сучасні пошукові системи звертають особливу увагу на технічний стан веб-сайту. Якщо сайт заражений вірусом або його сторінки завантажуються повільно через використання застарілих технологій, не можна очікувати високих позицій у результатах пошуку;

– без ефективної навігації дизайн веб-сайту неможливий. Вона визначає, як клієнти дізнаються про вас та вашу компанію. Навігація повинна бути яскравою та легкою у використанні. Видимі посилання на домашню сторінку ("Home"), кнопки «About» і «Back» є кількома прикладами;

– рекомендується розпочинати розробку сайту з аналізу конкурентів та складання плану пошукової оптимізації (SEO). У світі існує величезна кількість веб-сайтів, з якими новим сайтам доведеться змагатися в пошуковій видачі. Щоб спростити просування сайту, бажано визначити ключові слова, за якими проводитиметься просування ще до початку розробки сайту. Сучасні алгоритми пошукових систем більше не керуються заздалегідь заданими формулами, створеними людьми. Формула ранжування для кожного запиту та кожного сайту визначається пошуковою системою у процесі аналізу конкурентів за кожним ключовим словом. Тому дуже важливо вивчити та ретельно проаналізувати ключові показники сайтів конкурентів ще до початку роботи над власним сайтом.

3 АНАЛІЗ АНАЛОГІВ

Аналіз конкурентів є важливим етапом розробки сайту. Цей процес допомагає розробникам отримати інформацію про інші сайти, які пропонують аналогічні послуги або продукти, та зрозуміти їх переваги, слабкі місця та загальну стратегію.

Ось кілька ключових аспектів аналізу конкурентів, які можуть бути корисними при розробці сайту:

- ідентифікація конкурентів. Спочатку потрібно знайти конкурентів, які продають схожі товари чи послуги онлайн. Це можуть бути як прямі конкуренти – ті, які пропонують такі самі продукти або послуги, так і непрямі конкуренти – ті, які можуть задовольнити ту ж потребу в аудиторії;

- аналіз дизайну та користувальницького досвіду. Вивчити дизайн та користувальницький досвід (UX) конкуруючих сайтів. Оцініть їх візуальний дизайн, навігацію, розташування елементів та зручність використання. Визначить, що працює добре, і що можна покращити, щоб створити більш привабливий та зручний сайт;

- зміст та функціональність. Проаналізувати зміст та функціональність конкуруючих сайтів. Вивчити, які види інформації надаються, які функції доступні, як вони представлені, та як вони взаємодіють із користувачем. Визначте основні аспекти, які варто взяти до уваги під час розробки веб-сайту;

- маркетингові стратегії. Вивчіть маркетингові стратегії конкурентів. Дослідити, як вони залучають трафік на свої сайти, яку рекламу використовують, які канали маркетингу активно застосовують. Це дозволить визначити успішні підходи та ідеї для просування сайту.

- відгуки та репутація. Зверніть увагу на відгуки користувачів і репутацію конкурентних сайтів. Вивчіть думки клієнтів щодо товарів, послуг і загального досвіду роботи. Це може допомогти знайти переваги та проблеми, які можна покращити;

– треба бути в курсі інновацій та тенденцій у галузі. Які нові функції, технології чи підходи застосовують конкуруючі сайти. Це допоможе визначити, які нововведення можуть бути включені на сайті для залучення та утримання аудиторії.

Розглянемо кілька сайтів.

"Smachno" (<https://smachno.ua/>) – це український кулінарний портал, на якому можна знайти безліч рецептів української кухні. Сайт пропонує широкий вибір страв, починаючи від традиційних українських супів та інших страв до випічки, десертів та напоїв. Smachno.ua має свої переваги та недоліки.

Переваги:

– великий вибір рецептів. Сайт пропонує велику колекцію рецептів української кухні, що робить його привабливим для відвідувачів, які шукають різноманітність та різноманітні варіанти страв;

– інтерфейс користувача. Smachno.ua має зручний інтерфейс, який легко використовувати. Це важливо для користувачів, які люблять легко шукати рецепти та переміщатися по сайту;

– фотографії та описи. Кожен рецепт супроводжується якісними фотографіями страви та докладними описами, що робить сайт привабливим та інформативним для користувачів;

– категоризація та пошук. Smachno.ua пропонує зручне категорювання рецептів відповідно до типів страв, інгредієнтів і складності приготування. Крім того, сайт пропонує функцію пошуку, яка полегшує користувачам пошук рецептів;

– взаємодія з користувачами. Користувачі можуть оцінювати та коментувати рецепти, що сприяє взаємодії та обміну думками серед спільноти.

Недоліки:

– дизайн. Візуальний дизайн Smachno.ua можна вважати застарілим і несучасним. Необхідно оновити веб-сайт, щоб відповідати сучасним вимогам дизайну та візуального досвіду користувачів;

– мобільна адаптація. Деякі користувачі можуть мати проблеми з переглядом Smachno.ua на мобільних пристроях, оскільки сайт може не бути повністю оптимізований для мобільної адаптації. Це може знижувати зручність використання і призводити до погіршення користувальницького досвіду.

AllRecipes (<https://www.allrecipes.com/>) – це один із найпопулярніших сайтів з рецептами. Тут можна знайти величезну базу рецептів різних кухонь світу від простих щоденних страв до вишуканих рецептів для особливих випадків.

Переваги:

– у AllRecipes є велика колекція рецептів різних кухонь по всьому світу. Це робить його цінним ресурсом для людей, які шукають варіативність і різноманітність у кулінарних ідеях;

– користувачі можуть створювати свої акаунти на AllRecipes, ділитися своїми власними рецептами і залишати коментарі та оцінки до інших рецептів. Це сприяє створенню активної спільноти та взаємодії між користувачами;

– різноманітні функції. AllRecipes пропонує різні функції, можна використовувати AllRecipes для створення списків покупок, збереження рецептів у своєму особистому кабінеті та багато іншого. Це дозволяє клієнтам покращити свій досвід використання веб-сайту;

– сайт має зручний та інтуїтивно зрозумілий інтерфейс, який дозволяє користувачам легко знайти потрібні рецепти, переглядати кроки приготування та знаходити додаткові інформації про продукти та кулінарні прийоми.

Недоліки:

– AllRecipes отримує фінансування від реклами. У деяких випадках велика кількість рекламних банерів і спливаючих вікон може дратувати користувачів і негативно впливати на загальний досвід користувача;

– обсяг сторінок. Через велику кількість інформації та зображень на сторінках сайту, завантаження сторінок може займати більше часу, особливо

при повільному інтернет-з'єднанні. Це може бути неприємним для користувачів, які бажають швидко знайти потрібний рецепт;

- навігація та пошук. Користувачі можуть вважати систему пошуку та навігації сайту трохи незрозумілою. Можливо, веб-сайт міг би покращити функціональність пошуку та структуру категорій, щоб зробити більш простим для користувачів знайти потрібну інформацію.;

- у зв'язку з тим, що AllRecipes надає можливість користувачам вносити свої рецепти, якість деяких рецептів може змінюватись. Деякі рецепти можуть бути неправильними, недостатньо докладними або не відповідати очікуванням користувачів.

Food Network (<https://www.foodnetwork.com/>) – це сайт, що пропонує рецепти від відомих шеф-кухарів та кулінарних експертів. Тут можна знайти різноманітні рецепти, відеоуроки та поради щодо приготування.

Переваги:

- Food Network пропонує рецепти, відеоуроки та статті від професійних кухарів і кулінарних експертів. Це створює довіру та приваблює споживачів, які шукають перевірені та якісні рецепти;

- сайт пропонує широкий вибір відеоуроків для приготування страв різних кухонь світу. Відеоінструкції полегшують процес приготування та дозволяють користувачам краще зрозуміти техніки та прийоми кулінарії;

- Food Network створює власні рецепти та контент, що робить його унікальним та відрізняє від інших сайтів з рецептами. Це приваблює користувачів, які шукають нові та оригінальні ідеї для приготування страв;

- проста навігація та пошук. Сайт має ефективну систему пошуку та навігації, а його дизайн простий для розуміння. Це полегшує користувачам швидкий доступ до рецептів, відео та іншого контенту.

Недоліки:

- обмежений доступ до певного контенту. Деякі відеоуроки та статті можуть бути доступні лише для платних передплатників. Це може бути

розчаруванням для користувачів, які чекають на безкоштовний доступ до всіх контентів;

- Food Network, як і багато інших веб-сайтів, фінансується рекламою.

Іноді реклама може бути надмірною та відволікати користувачів;

- не завжди докладні інструкції. Деякі рецепти на Food Network можуть бути короткими, або не містити достатньо докладних інструкцій. Це може ускладнити розуміння процесу приготування для деяких користувачів, особливо для новачків;

- мало взаємодії з користувачами. На відміну від деяких інших веб-сайтів, Food Network не надає користувачам можливості коментування та оцінювання рецептів. Це обмежує досвід і взаємодію користувачів.

4 ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ

Існує безліч засобів розробки веб-сайтів і вибір певного інструменту залежить від вимог проекту, навичок розробника та інших факторів. Ось кілька популярних засобів розробки веб-сайтів:

- HTML та CSS. HTML (мова гіпертекстової розмітки) та CSS (каскадні таблиці стилів) є основними мовами для створення структури та візуального оформлення веб-сторінок. Вони є основою будь-якого веб-сайту;

- JavaScript. JavaScript є мовою програмування, яка дозволяє створювати інтерактивні елементи на веб-сторінках. Вона широко використовується для розробки динамічних функцій, валідації форм, створення анімації тощо;

- фреймворки. Фреймворки, такі як React, Angular та Vue.js, пропонують інструменти та бібліотеки, які полегшують створення веб-додатків. Вони дозволяють обробляти стан програми та створювати складні інтерфейси;

- середовище розробки. Середовища розробки, такі як Visual Studio Code, Sublime Text, Atom та інші, пропонують зручне середовище для написання коду. Вони забезпечують підсвічування синтаксису, автодоповнення коду, налагодження та інші корисні функції;

- бази даних, такі як MySQL, PostgreSQL та MongoDB, використовуються для збору даних, таких як інформація про користувачів, контент та інші;

- системи керування версіями. Системи керування версіями, такі як Git, дозволяють відстежувати зміни в коді та співпрацювати з іншими розробниками. Вони також забезпечують контроль версій та відновлення попередніх версій коду;

- інструменти тестування. Для забезпечення якості коду та функціональності веб-сайту використовуються інструменти тестування, такі

як Selenium, Jest, Mocha та інші. Вони дозволяють автоматизувати тестування та перевірку працездатності веб-сайту;

– засоби розгортання. Розгортання веб-сайту на сервері може відбуватися за допомогою багатьох інструментів, таких як Docker, Heroku, AWS і Netlify. Вони гарантують, що веб-сайт буде доступним і працездатним, що полегшує процес розміщення веб-сайту.

Крім перерахованих вище засобів, існують інші інструменти та технології, які можуть бути використані при розробці веб-сайту в залежності від конкретних потреб проекту. Важливо вибрати інструменти, які найкраще відповідають вимогам проекту і якими розробник володіє або готовий освоїти.

Основні етапи розробки веб-сайту включають проектування, розробку інтерфейсу, програмування функціональності, тестування, оптимізацію та розгортання на сервері. Кожен із цих етапів може вимагати використання різних засобів розробки веб-сайтів.

Розробка веб-сайтів – це ітеративний процес, який потребує постійного розвитку та покращення. Розробники повинні бути готові вивчати нові інструменти та технології, щоб залишатися в курсі останніх тенденцій і гарантувати, що їхні проекти будуть виконані з високою якістю.

Давайте детально розглянемо деякі із цих засобів.

4.1 Розробка HTML структури

HTML (HyperText Markup Language) – це мова гіпертекстової розмітки, яка використовується для створення структури та вмісту веб-сторінок. HTML є основою для створення веб-сайтів і використовується в поєднанні з CSS і JavaScript для створення повнофункціональних веб-додатків.

Структура HTML документа ґрунтується на елементах, які описують різні частини сторінки. Ось деякі основні елементи HTML:

– теги, що відкривають і закривають, використовуються для обрамлення HTML-елементів. Наприклад, абзаци позначаються тегом <p>,

тоді як заголовки першого рівня позначаються тегом `<h1>`. Назву елемента містить відкриваючий тег, а закривальний тег починається з символу `/`, за яким слідує назва елемента;

- атрибути. Атрибути використовуються для додаткової інформації про елемент. Вони вказуються у тезі елемента і зазвичай мають ім'я і значення. Наприклад, атрибут `href` у тегу `<a>` вказує URL-адресу посилання;

- елементи контенту. HTML підтримує різні елементи для розміщення контенту на веб-сторінці. Деякі з них включають `<p>` для абзаців, `<h1>`-`<h6>` для заголовків різних рівнів, `` та `` для списків, `` для вставки зображень тощо;

- структурні елементи. HTML містить структурні елементи, які визначають різні частини веб-сторінки. Наприклад, `<header>` використовується для верхньої частини сторінки, `<nav>` для навігаційного меню, `<main>` для основного вмісту сторінки, `<footer>` для нижньої частини сторінки тощо;

- форми. HTML надає елементи форми, які дозволяють користувачам вводити дані. Це можуть бути текстові поля, прапорці, радіокнопки, списки та інші. Форми визначаються з допомогою тега `<form>`;

- посилання. HTML дозволяє створювати гіперпосилання за допомогою тега `<a>`. Атрибут `href` вказує на адресу, на яку буде здійснюватися перехід при натисканні на посилання. Також можна використовувати атрибут `target` для вказівки, як буде відкрито посилання (у поточному вікні, новій вкладці);

- вбудовані елементи. HTML підтримує вбудовування різних медіа-елементів, таких як зображення, відео та аудіо. Наприклад, тег `` використовується для вбудовування зображень, а тег `<video>` для вбудовування відео.

Крім перелічених вище елементів, HTML має безліч інших можливостей, таких як створення таблиць за допомогою `<table>`, використання стилів за допомогою атрибуту `style`, вставка скриптів за допомогою тега `<script>`.

HTML документ складається з оголошення типу документа `<!DOCTYPE html>`, елемента `<html>`, що містить заголовок документа `<head>` та тіло документа `<body>`. Всередині `<head>` можна вказати заголовок сторінки, мета-теги, стилі CSS та посилання на скрипти.

Всі елементи HTML можуть бути стилізовані за допомогою CSS для завдання зовнішнього вигляду сторінки, а також можуть бути пов'язані з JavaScript для додавання інтерактивності та динамічної поведінки. Існують більш складні концепції, такі як розмітка таблиць, форматування тексту та робота з мультимедіа, які також можуть бути вивчені для створення більш сучасних веб-сайтів і додатків.

4.2 Розробка CSS стилізація

CSS (Cascading Style Sheets) – це мова стилів, що використовується для визначення зовнішнього вигляду та форматування елементів HTML на веб-сторінці. За допомогою CSS можна керувати кольорами, шрифтами, розмірами, розташуванням елементів та іншими аспектами дизайну веб-сайту.

CSS працює за принципом каскаду, що означає, що стилі можуть успадковуватись від батьківських елементів та перевизначатися більш специфічними стилями. Це дозволяє створювати гнучкі та масштабовані стилі для всього веб-сайту.

Ось основні концепції та можливості CSS:

- селектори. Селектори CSS використовуються для вибору елементів, до яких будуть застосовані стилі. Наприклад, селектор `h1` вибирає всі елементи першого рівня, а селектор `.classname` вибирає всі елементи з певним класом;

- CSS визначає властивості, які можуть бути застосовані до елементів, та значення, які встановлюються для цих властивостей. Наприклад, властивість `color` керує кольором тексту, а властивість `font-size` встановлює розмір шрифту;

- каскадування та успадкування. Каскадування CSS дозволяє визначати стилі на різних рівнях (наприклад, усередині файлу стилів,

всередині блоку <style> або прямо в HTML елементі) і об'єднувати їх для застосування до елементів. Спадкування дозволяє властивостям успадковуватись від батьківських елементів, що спрощує застосування загальних стилів до групи елементів;

- класи та ідентифікатори. Класи та ідентифікатори дозволяють створювати більш специфічні селектори та застосовувати стилі до певних елементів. Класи вказуються за допомогою точки (.), а ідентифікатори – за допомогою решітки (#);

- бокс модель. Кожен елемент у HTML має бокс модель, який визначає його розміри, відступи, межі та відступи. За допомогою CSS можна контролювати ці аспекти дизайну, використовуючи властивості, такі як `width`, `height`, `padding`, `border` та `margin`;

- позиціонування. CSS надає різні способи позиціонування елементів на сторінці. Деякі з них включають `static`, `relative`, `absolute` та `fixed`. Ці властивості дозволяють контролювати положення елементів щодо інших елементів чи вікна перегляду;

- анімація та переходи. За допомогою CSS можна створювати анімацію та переходи для елементів, додаючи плавні ефекти та зміни властивостей з часом. Це досягається за допомогою властивостей, таких як `animation` та `transition`, які дозволяють встановлювати ключові кадри та часові інтервали анімації;

- медіа запити. За допомогою медіа-запитів можна адаптувати стилі залежно від розміру екрана або пристрою, на якому відображається веб-сторінка. Це особливо важливо для створення чуйного дизайну, який адаптується до різних роздільних здатностей екрану.

CSS має безліч властивостей, функцій та можливостей, які дозволяють створювати різноманітні та красиві дизайни веб-сайтів. При розробці важливо вивчити різні концепції CSS та експериментувати з ними, щоб досягти бажаного вигляду для вашого веб-сайту.

4.3 Препроцесор CSS

SCSS (Sassy CSS) – це препроцесор CSS, який розширює функціональність звичайного CSS, надаючи розробникам потужніші інструменти для роботи зі стилями та організації коду. SCSS є частиною сімейства препроцесорів, які дозволяють використовувати змінні, міксини, вкладені правила та інші можливості, що спрощують та покращують процес розробки CSS.

Докладний опис основних можливостей SCSS:

- SCSS дозволяє визначати змінні для використання у CSS. Змінні можуть містити значення кольорів, розміри шрифтів, відступи та інші стильові властивості. Це спрощує та прискорює процес зміни стилів, оскільки зміна значення змінної автоматично відбивається на всіх місцях, де вона використовується;

- вкладені правила. SCSS дозволяє вкладати одне правило в інше, спрощуючи організацію коду та підвищуючи його читабельність. Вкладені правила автоматично успадковують стилі батьківського елемента, що усуває необхідність повторного вказівки селекторів;

- міксини SCSS дозволяють визначати групи стилів, які можуть бути повторно використані в різних частинах коду. Міксини можуть приймати аргументи і використовуватися для створення різних стилів залежно від переданих значень;

- SCSS надає набір вбудованих функцій, які полегшують роботу зі значеннями властивостей, такими як кольори, розміри та одиниці виміру. Ці функції можуть бути використані для виконання обчислень або перетворення значень;

- SCSS підтримує можливість імпортувати та об'єднувати різні SCSS файли, що спрощує організацію та управління стилями проекту. Це дозволяє розділити код на дрібніші модулі та повторно використовувати їх у різних частинах проекту;

- спадкування. SCSS підтримує успадкування стилів за допомогою директиви `@extend`. Це дозволяє створювати базові стилі та розширювати їх в інших класах чи селекторах, зменшуючи дублювання коду та покращуючи його супроводжуваність;

- умовні оператори та цикли. SCSS підтримує умовні оператори (`if-else`) та цикли, що дозволяє створювати більш гнучкі стилі, які залежать від умов або потребують ітерації за набором значень.

4.4 Мова програмування JavaScript

JavaScript – це високорівнева мова програмування, яка широко використовується для розробки динамічних веб-сайтів і веб-додатків. Вона дозволяє додавати інтерактивність, обробку подій, маніпуляцію з елементами сторінки та багато іншого.

Ось основні концепції та можливості JavaScript:

- JavaScript має синтаксис, схожий на інші мови програмування, такі як C++ і Java. Він використовує змінні для зберігання даних, оператори для виконання операцій, умовні оператори для прийняття рішень та цикли для повторення операцій;

- типи даних. JavaScript підтримує різні типи даних, включаючи числа, рядки, логічні значення (`true/false`), об'єкти, масиви та інші. Динамічна типізація дозволяє змінним змінювати свій тип у процесі виконання програми;

- об'єкти. JavaScript базується на об'єктній моделі, і все в JavaScript є об'єктами або може бути представлено у вигляді об'єктів. Об'єкти являють собою набір якостей та способів, які можуть бути використані для маніпуляції даними та виконання операцій;

- функції. Функції JavaScript є основними будівельними блоками організації коду. Вони можуть бути визначені як іменовані функції або анонімні функції, та можуть приймати аргументи та повертати значення. Функції може бути викликані до виконання певних операцій чи обробки подій;

– JavaScript дозволяє обробляти події, такі як клацання миші, натискання клавіш, завантаження сторінки та інші. Події можуть бути пов'язані з певними елементами на сторінці і JavaScript дозволяє реагувати на події, та виконувати відповідні дії;

– маніпуляція DOM. DOM (Document Object Model) є структурою HTML документа за допомогою дерева об'єктів. JavaScript дозволяє отримувати доступ до елементів сторінки через DOM, змінювати їх зміст, стилі та атрибути, додавати або видаляти елементи, і таким чином динамічно змінювати зовнішній вигляд та поведінку сторінки;

– AJAX (Asynchronous JavaScript and XML) дозволяє обмінюватися даними між браузером та сервером без перезавантаження сторінки. Це дозволяє створювати більш інтерактивні та чуйні веб-програми, які можуть асинхронно оновлювати зміст сторінки;

– JavaScript Object Notation (JSON) – це формат для представлення структурованих даних у вигляді об'єктів JavaScript. JSON дозволяє передавати дані між клієнтом та сервером у зручному та компактному форматі;

– модулі та бібліотеки. JavaScript підтримує модульність, що дозволяє розділяти код на окремі модулі та перевикористувати його в різних частинах програми. Є також безліч бібліотек та фреймворків, таких як React, Angular та Vue.js, які спрощують розробку складних веб-додатків.

JavaScript є потужною та гнучкою мовою програмування, яка широко використовується для розробки веб-сайтів та додатків. Вона надає безліч можливостей для створення інтерактивних і динамічних інтерфейсів користувача.

4.5 Бібліотека jQuery

jQuery – це швидка, легковажна, та потужна бібліотека JavaScript, яка спрощує роботу з DOM (Document Object Model), обробку подій, анімацію та виконання AJAX запитів. Вона надає зручний та крос-браузерний спосіб взаємодії з елементами веб-сторінки та управління ними.

Основні концепції та можливості jQuery:

- вибір елементів. Однією з основних можливостей jQuery є зручний вибір елементів на сторінці за допомогою селекторів CSS. Селектори jQuery дозволяють вибирати елементи за їх ідентифікатором (`#id`), класом (`.class`), типом елемента (`element`), атрибутами та іншими критеріями;
- маніпуляція елементами. jQuery надає безліч методів маніпуляції елементами на сторінці. З їх допомогою можна змінювати вміст елементів, додавати або видаляти елементи, змінювати атрибути, стилі та багато іншого. Наприклад, метод `.text()` дозволяє встановити або отримати текстовий вміст елемента, а метод `.addClass()` дозволяє додати клас до елемента;
- обробка подій. jQuery спрощує обробку подій, таких як клацання миші, натискання клавіш, наведення курсору та інші. Методи jQuery, такі як `.click()`, `.keydown()`, `.hover()` та багато інших, дозволяють призначати обробники подій та реагувати на них за допомогою функцій зворотного виклику;
- анімація. jQuery надає безліч методів для створення анімацій на веб-сторінці. З їх допомогою можна анімувати зміни властивостей елементів, таких як розмір, позиція, прозорість та інші. Наприклад, метод `.animate()` дозволяє плавно змінювати властивості елемента із заданою тривалістю та ефектами;
- jQuery надає зручні методи для виконання AJAX запитів та роботи з даними на сервері. Методи `.ajax()` та `.get()` дозволяють отримувати дані з сервера без перезавантаження сторінки, надсилати дані на сервер та виконувати інші операції з використанням AJAX;
- ефекти та плагіни. jQuery має безліч вбудованих ефектів, таких як приховування, показ, плавна поява та зникнення елементів. Крім того, існує безліч плагінів, розроблених спільнотою, які розширюють можливості jQuery, додаючи нові функції та компоненти;
- утиліти та зручності. jQuery пропонує різні утиліти та функції для полегшення роботи з JavaScript. Наприклад, метод `.each()` дозволяє

перебирати елементи та виконувати операції над ними, метод `.extend()` використовується для розширення об'єктів та функцій, а метод `.ajax()` спрощує виконання AJAX запитів.

jQuery була популярна і широко використовувана бібліотека JavaScript протягом багатьох років, але з розвитком сучасних фреймворків і можливостей мови, її популярність трохи знизилася. Тим не менш, вона залишається корисним інструментом для швидкої та простої розробки веб-застосунків.

4.6 Система управління версіями Git

Система управління версіями Git є однією з найпопулярніших і найпотужніших систем управління версіями, що широко використовуються розробниками по всьому світу. Git був створений для управління розробкою ядра Linux, і з того часу він став стандартним інструментом розробки програмного забезпечення.

Основні концепції та можливості Git:

- репозиторій. Git працює з репозиторієм, який містить усі версії файлів та історію змін. Репозиторій може бути локальним чи віддаленим. Локальний репозиторій знаходиться на комп'ютері розробника, а віддалений репозиторій зазвичай зберігається на сервері і використовується для спільної роботи та синхронізації змін між різними розробниками;

- коміти. Git зберігає зміни у вигляді комітів. Коміт є знімок стану репозиторію в певний момент часу. Кожен коміт має унікальний ідентифікатор і містить інформацію про внесені зміни, автора, часу та інших метаданих;

- гілки. Git створює та працює з гілками. Гілка є окремою лінією розробки, яка може містити свої зміни та коміти. Можна паралельно розробляти різні функціональності чи виправлення помилок, використовуючи гілки, не впливаючи на основну гілку розробки. Крім того, гілки полегшують злиття змін між різними гілками різну;

– злиття та конфлікти. Git надає механізми для злиття змін із різних гілок. При злитті Git автоматично поєднує зміни, якщо вони не конфліктують. Однак, якщо Git виявляє конфлікт змін, які не можуть бути автоматично об'єднані, він позначає ці місця як конфлікти та потребує ручного вирішення;

– підновлення та повернення змін. Git надає можливість відкотитися до попередніх комітів або відновити видалені файли. Це корисно, коли необхідно виправити помилки, або повернутися до попереднього стану проекту;

– видалені репозиторії та спільна робота. Можна синхронізувати локальні репозиторії з віддаленими репозиторіями на сервері за допомогою Git. Це дозволяє розробникам обмінюватися змінами, працювати разом і спостерігати за тим, як змінюються інші члени команди;

– інструменти та команди. Git має командний рядок, який дозволяє виконувати різні речі, такі як ініціалізація репозиторії, створення гілок, злиття змін і відкат, серед інших функцій. Крім того, для спрощення використання Git доступні інтегровані середовища розробки та графічні інтерфейси.

Git є потужним інструментом управління версіями, що надає розробникам гнучкість, швидкість та надійність під час роботи з проектами будь-якого масштабу. Він дозволяє ефективно управляти змінами, спільно працювати з іншими розробниками та відновлюватись від помилок.

5 РОЗРОБКА ПРАКТИЧНОЇ ЧАСТИНИ

5.1 Реалізація задач та мети

Метою створення сайту про національну кухню України є популяризувати українську кухню, підвищити інтерес до неї, залучити туристів, які подорожують країною, або завітали до України з-за кордону. Реклама українських ресторанів та кафе, українські страви можна приготувати самому на своїй кухні, або ж відвідати популярні ресторани або кафе. Сайт надає інформацію про національну кухню України, включаючи її історію, традиції, особливості регіональних кулінарних стилів, популярні інгредієнти та спеції, а також сезонні страви. Відвідувачі сайту мають змогу дізнатися про національні гастрономічні символи, типові методи приготування та популярні рецепти. Сайт пропонує широкий вибір рецептів українських страв із покроковими інструкціями та фотографіями. Також включені поради щодо вибору інгредієнтів, приготування страв та збереження їхньої автентичності. Відвідувачі мають змогу вивчати та експериментувати з українською кухнею у своїй власній кухні.

Ціллю створеного сайту є збереження та передача української кулінарної спадщини наступним поколінням. Він розповідає про значні культурні події, пов'язані з харчуванням, пропонуватиме рецепти по усіх регіонах країни а також, приділяє увагу традиційним методам приготування та використання інгредієнтів.

5.2 Інструментальні засоби розробки сайту

Figma – це інструмент для дизайну інтерфейсів, який широко використовується у веб-розробці для створення та прототипування дизайну сайтів. У контексті розробки дизайну для сайту про національну кухню України

Figma може бути дуже корисним інструментом. Figma дозволяє створювати детальні макети візуального оформлення сайту. Вона пропонує широкий набір інструментів для малювання, додавання тексту, зображень, іконок та інших елементів дизайну. За допомогою Figma можна створювати сторінки сайту, компоненти інтерфейсу, навігаційні елементи, форми та багато іншого.

Adobe Photoshop – це професійне програмне забезпечення для редагування та створення графічних зображень. За допомогою Photoshop можна редагувати та покращувати фотографії, які будуть використовуватися на сайті. Можна коригувати яскравість, контраст, насиченість, ретушувати недоліки та застосовувати різні ефекти, щоб створити привабливі та апетитні зображення страв національної кухні. Після створення дизайну та графічних елементів у Photoshop можна експортувати їх до різних форматів, таких як PNG, JPEG або SVG, щоб використовувати їх на сайті. Це забезпечить оптимізацію, та відповідність вимогам для швидкого завантаження та відображення на сайті.

Visual Studio Code – це популярний редактор коду від Microsoft, який є безкоштовним і широко використовується веб-розробниками та верстальниками різного рівня. Цей редактор пропонує широкі можливості і відмінно підходить як для початківців, так і для досвідчених розробників. З одного боку, VS Code приваблює новачків завдяки своєму інтуїтивно зрозумілому і простому інтерфейсу. Це дозволяє швидко освоїтися з редактором і почати працювати з кодом без складнощів. З іншого боку, в VS Code є багато функціональних можливостей, які привертають досвідчених розробників. Редактор підтримує розширення, які дозволяють налаштувати середовище розробки під власні потреби. Крім того, він пропонує різноманітні інструменти для швидкої навігації, редагування і відлагодження коду, що допомагає ефективно працювати з проектами будь-якої складності. В загальному, Visual Studio Code є потужним і гнучким редактором, який задовольняє потреби як початківців, так і досвідчених розробників, спрощуючи роботу з кодом і покращуючи продуктивність.

5.3 Проектування структури сайту

Багатосторінкові сайти мають переваги порівняно зі статичними односторінковими сайтами. Вони дозволяють створювати більший контент і структурувати інформацію за допомогою різних сторінок. Багатосторінкові сайти дозволяють використовувати різноманітні функції і інтерактивні елементи, такі як форми зворотного зв'язку, коментарі, пошукові поля, галереї зображень та інше. Це дає можливість створювати більш розширені та функціональні веб-додатки. Такий сайт має деякі переваги:

- краща структура і навігація. Багатосторінкові сайти дозволяють розділити інформацію на різні сторінки зі зрозумілою структурою. Це полегшує навігацію для відвідувачів і допомагає їм швидше знайти потрібну інформацію;

- краща оптимізація для пошукових систем. Багатосторінковий сайт може містити більше ключових слів, що позитивно впливає на SEO. Кожна сторінка може бути оптимізована для конкретного ключового слова або тематики, що збільшує ймовірність, що сайт з'явиться у результатах пошуку;

- легша розширюваність. Багатосторінкові сайти мають більшу гнучкість при додаванні нового контенту або функціональності. Вони можуть легко розширюватись шляхом додавання нових сторінок або розділів, що дозволяє гнучко пристосовувати сайт до змінних потреб аудиторії;

- більша гнучкість у редагуванні та оновленні: Багатосторінкові сайти надають більше можливостей для редагування та оновлення вмісту. Кожна сторінка може бути змінена окремо без впливу на інші сторінки. Це забезпечує зручність та ефективність у внесенні змін на сайті, дозволяючи результативно оновлювати інформацію та підтримувати актуальність сайту;

- розширення потенціалу дизайну. Багатосторінкові веб-сайти надають більшу свободу в дизайні, оскільки для кожної сторінки можна застосувати різні елементи, макети та стилі. Це дозволяє створювати більш різноманітні та привабливі дизайни, які відповідають особливостям кожної сторінки;

– покращена аналітика та відстеження даних. Багатосторінкові сайти надають більші можливості для аналізу та відстеження даних. Кожна сторінка може бути відслідкована окремо, що дозволяє отримувати детальну статистику про відвідуваність, поведінку користувачів та конверсії на кожній сторінці. Це допомагає зрозуміти, як ефективно працює сайт та як покращити його результативність.

Багатосторінкові сайти забезпечують більшу гнучкість, можливості та організованість в порівнянні зі статичними односторінковими сайтами. Вони дозволяють створювати більш повний та структурований контент, а також забезпечують кращі можливості для дизайну, SEO та редагування.

На початковому екрані буде розміщено вражаюче зображення, що відображає необхідну тематику, а також компактний текстовий блок, який має привернути увагу глядача до ключових моментів. Крім того, там буде розташована кнопка, на яку користувач зможе натиснути. Після чого він перейде до наступної частини головної сторінки, де він зможе обрати місто, кухня якого, його цікавить. Це допоможе зосередити увагу відвідувачів на найважливіших елементах сайту.

Головна функція даного сайту буде розташована наприкінці головної сторінки, там буде інтерактивна карта України, користувач зможе обрати абсолютно будь-який обласний центр, після натискання він зможе отримати коротку інформацію про цей регіон. Далі у користувача буде вибір перейти далі на сторінку цього регіону або вибрати інший регіон.

Наступна сторінка називатиметься "Міста" (Cities), це ще один спосіб користувача познайомитися з містами України. Там розташована сітка міст, де користувач зможе обрати будь-яке місто, пам'ятка якого, йому більше цього подобається. Користувач при натисканні на картку міста зможе перейти на його сторінку і там ознайомитися з рецептом місцевої національної кухні.

Остання сторінка називатиме "Про проект" (About). На цій сторінці користувача зможе познайомитися з інформацією о проекті. Та познайомитися з людиною, яка створила цей сайт.

Таким чином можна побачити, що цей сайт має 3 основних сторінки і двадцять п'ять додаткових сторінок з інформацією о національній кухні конкретного міста України.

5.4 Створення графічного дизайну

В першу чергу, при створенні дизайну була обрана кольорова палітра, що складається з кольорів, які відмінно підходять один одному. Колірна палітра цього дизайну включає яскраве і контрастне поєднання трьох класичних кольорів: чорного, білого і жовтого. Допоміжним кольором у нас є відтінок малинового, він використовується коли треба привернути уваги користувача, до певного елемента при наведенні. Як показано на рис. 5.1.

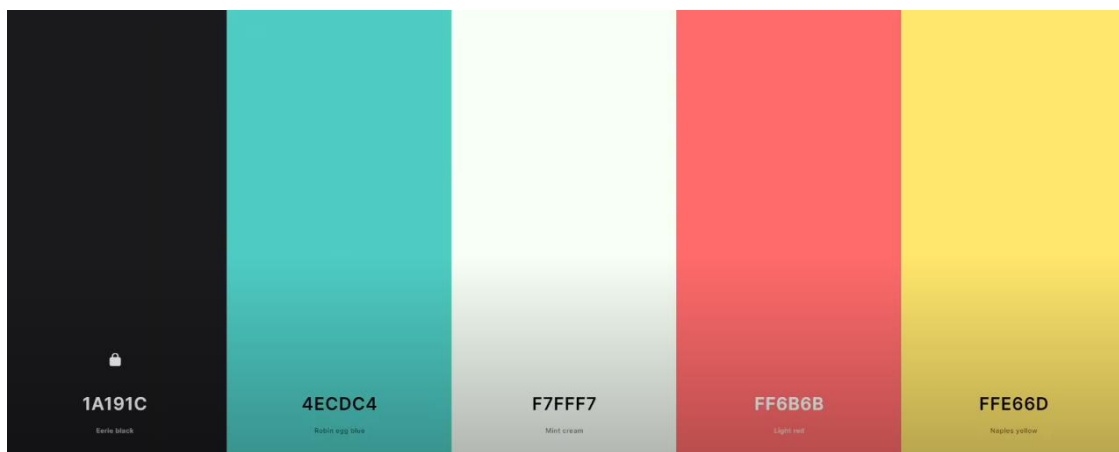


Рисунок 5.1 – Кольорова палітра

При розробці дизайну сайту був створений головний контейнер довжиною 1312 пікселів. Ця структура зберігається на усіх сторінках для того, щоб користувачам було легко сприймати інформацію на сайті. Для втілення дизайну сайту була обрана 12-колонова модульна сітка, оскільки вона є універсальною і надає широкі можливості для розміщення блоків та їх елементів.

На першій, головній сторінці (рис. 5.2) розташовується великий заголовок, який розкривається про що наш сайт, а також текстовий блок з

коротким описом. Праворуч розташоване яскраве та барвисте зображення. Також на цьому екрані є дві кнопки, з якими користувач може взаємодіяти, при натисканні на "Get Started", користувач перейде до наступного розділу сайту, інша кнопка "Watch Video", замінить зображені на відео.

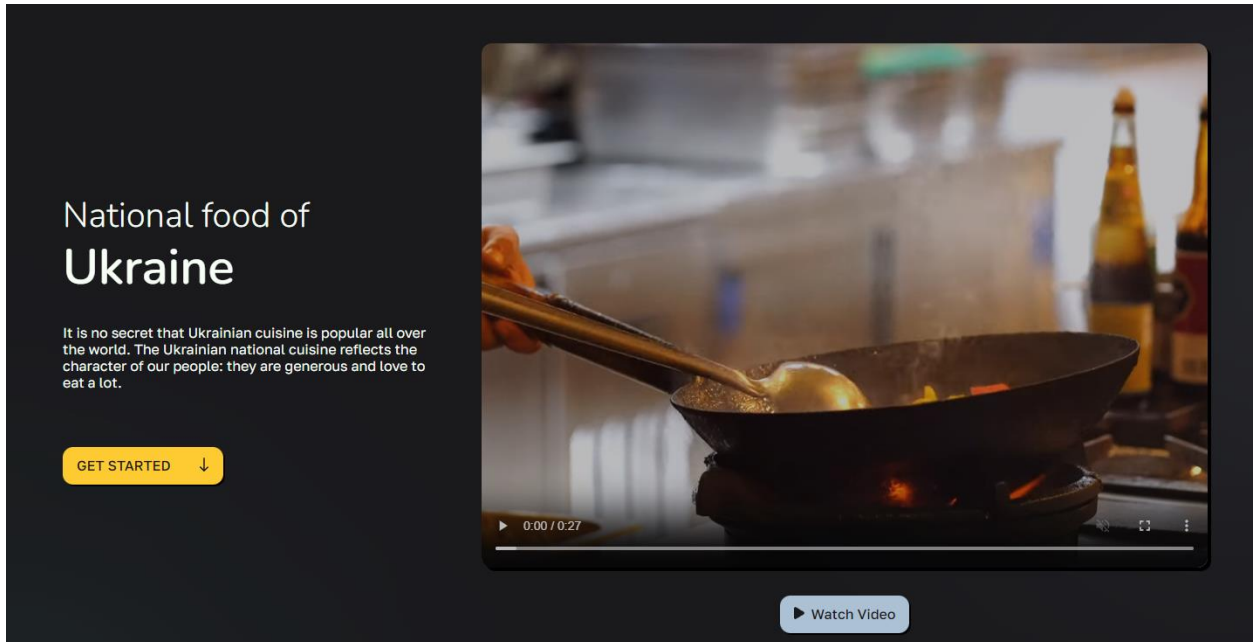


Рисунок 5.2 – Головна сторінка

На цій же головній сторінці, якщо перегорнути її вниз, розташовується наступний блок, в якому користувач отримує одну з трьох можливостей для вибору міста України, яке його цікавить. Також трохи нижче розташовується плашка з основними перевагами цього сайту.

На слайдері розташовуються великі міста, але якщо користувачеві цікаво побачити всі можливі варіанти, він може натиснути на кнопку і побачити наступний розділ з інтерактивною картою. При наведенні на неї мишкою, користувач побачить підказку з назвою міста, а при натисканні з'явиться вікно з коротким описом страви та фотографія, натиснувши на кнопку в цьому вікні, користувач перейде на сторінку цього міста. Розроблений дизайн сторінки представлений у додатку А на рис. А.1.

На другу сторінку можна перейти, натиснувши на неї, у верхній частині сайту. На цій сторінці міститься назва «Cities of Ukraine» (рис. 5.3).

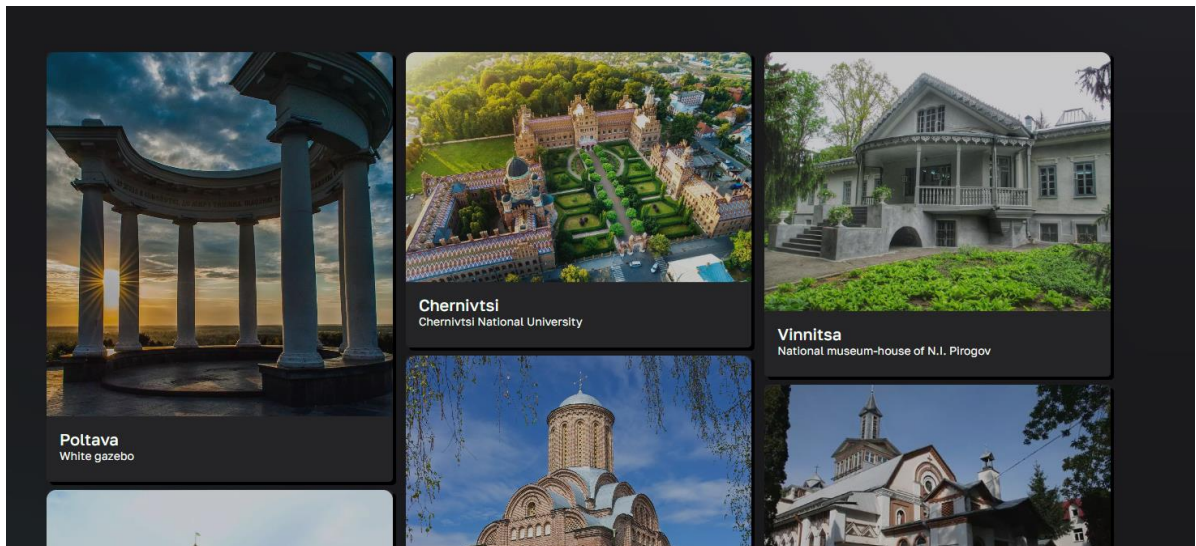


Рисунок 5.3 – Сторінка «Cities of Ukraine»

Також відразу ж користувач помітить плаваючі зображення з визначними пам'ятками великих міст України. Погортавши вниз користувач побачить картки з усіма містами України і зможе обрати, те місто, визначна пам'ятка якого, йому найбільше сподобалася. При натисканні користувач також перейде на сторінку з містом. Розроблений дизайн сторінки представлений у додатку А на рис. А.2.

На третій, останній сторінці користувач зможе побачити інформацію про те, хто і навіщо зробив цей сайт (рис. 5.4). На цю сторінку також можна перейти, натиснувши на неї, у верхній частині сайту. Розроблений дизайн сторінки представлений у додатку А на рис. А.3.

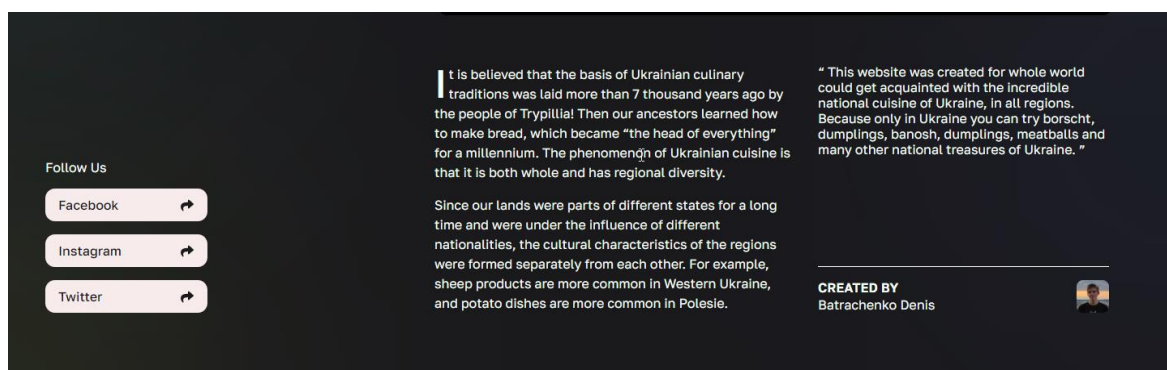


Рисунок 5.4 – Сторінка «About»

Сайт має двадцять п'ять сторінок по кожному місту України (рис. 5.5). Ці сторінки виконані в однаковому дизайні, який має два блоки. Перший блок розповість користувачеві про страву, яка є національним надбанням цього міста. На малюнку в цьому ж блоці користувач побачить, як вона виглядає.

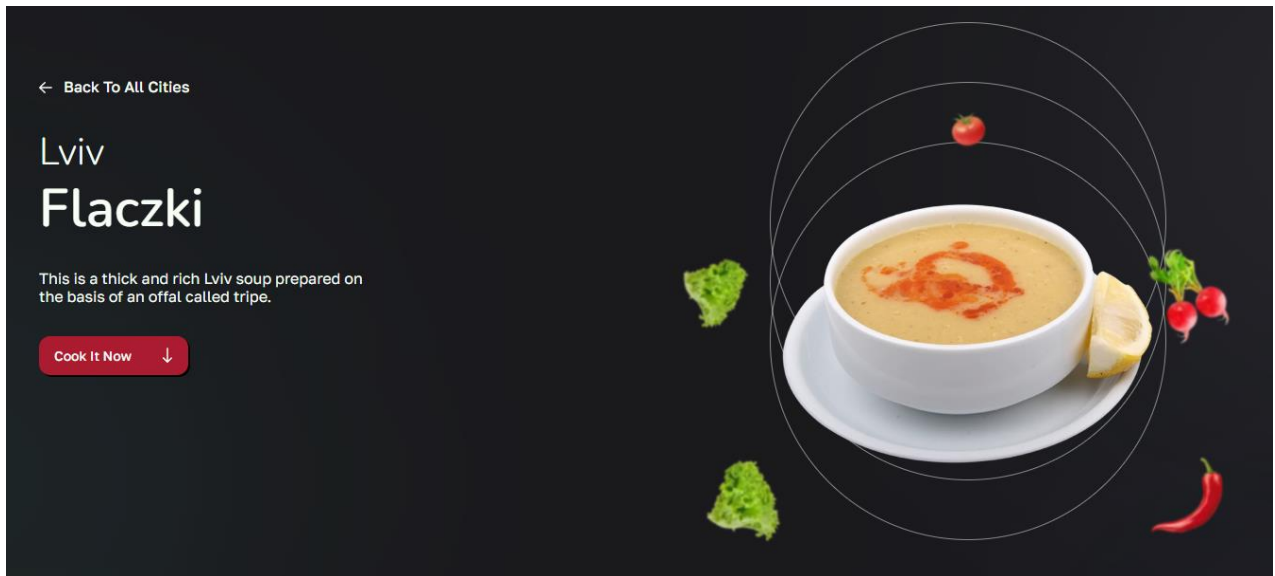


Рисунок 5.5 – Сторінка «Lviv»

Натиснувши на кнопку, користувач перейде до другого блоку (рис. 5.6). На ньому розміщений рецепт з усіма необхідними інгредієнтами та покроковою інструкцією до приготування цієї страви, а також час готування. Розроблений дизайн сторінки представлений у додатку А на рис. А.4.

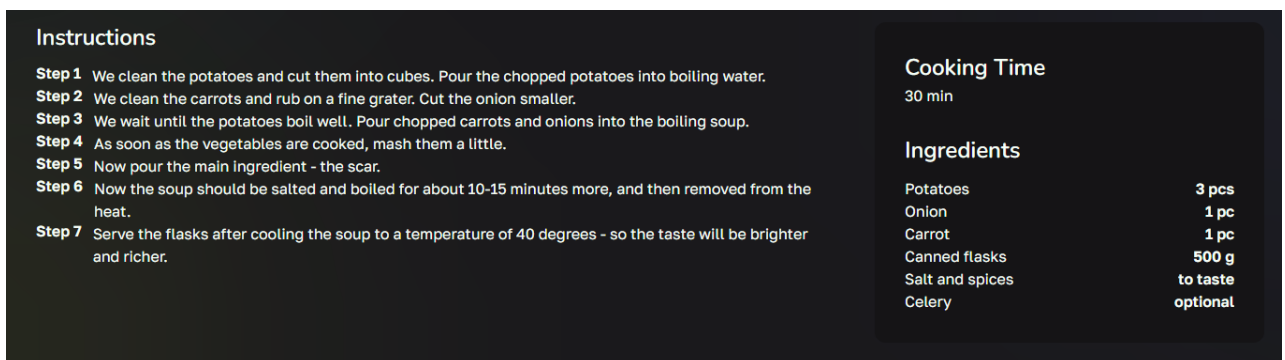


Рисунок 5.6 – Блок з інструкцією

5.5 Розміщення інформації у блоках

Усі блоки виконані за допомогою семантичних тегів. Таких як `header`, `nav`, `main`, `section`, `footer`.

`Header` – це елемент, що відкриває, зазвичай складається з набору вступних або навігаційних елементів. На цьому сайті він містить назву сайту, навігаційну панель представлених сторінок та список соціальних мереж.

Елемент `nav` використовується для визначення окремої секції документа, призначеної для навігаційних посилань. Ці посилання ведуть на усі представлені сторінки на сайті (рис. 5.7).

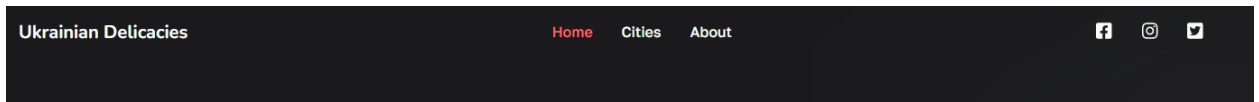


Рисунок 5.7 – Блок навігаційних посилань

Елемент `main` призначений для основного контенту (вмісту) документа (сторінки). Основний контент складається з контенту, що безпосередньо належить до головної теми документа або його розвиває.

Елемент `section` використовується для створення автономних розділів у HTML-документі. Він є самостійним блоком контенту, який не може бути більш точно визначений семантичними елементами. Зазвичай, але не завжди кожен розділ має свій заголовок, який допомагає ідентифікувати та описати вміст розділу. Таким чином більшість блоків на цьому сайті є секціями, які містять заголовок, короткий тестовий опис, та зображення (рис. 5.8).

Елемент `footer` використовується для створення нижнього колонтитулу (футера або підвалу) для найближчого секційного контенту або кореневого елемента секції. Футер містить інформацію про авторські права, посилання на соціальні мережі, та інформацію про автора (рис. 5.9).

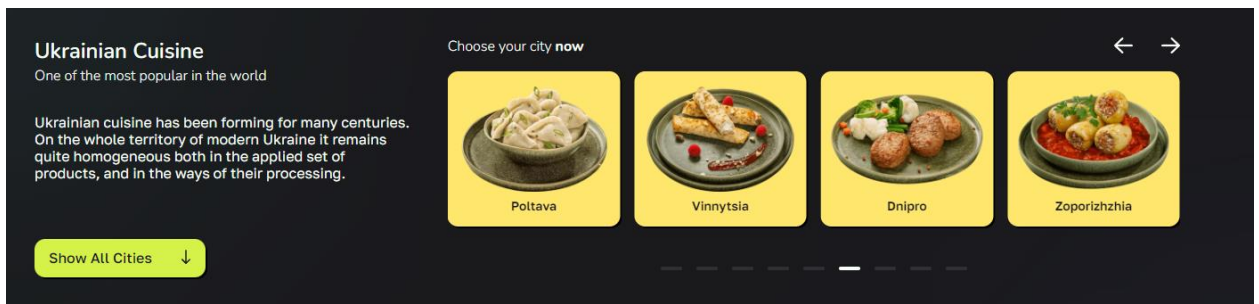


Рисунок 5.8 – Блок, який пропонує обрати місто

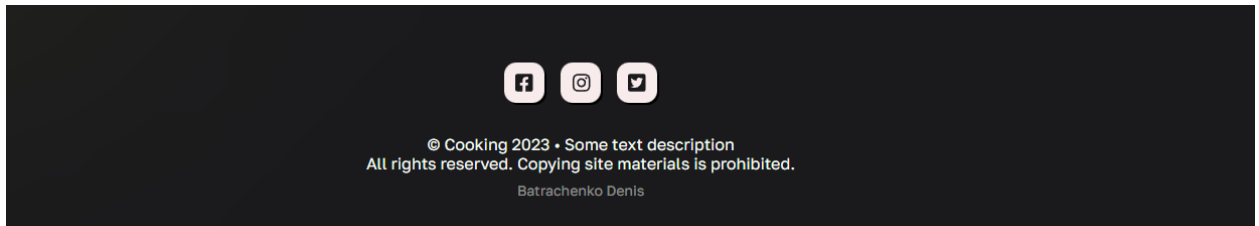


Рисунок 5.9 – Підвал

6 ЕКОНОМІЧНА ЧАСТИНА

В результаті виконання кваліфікаційної роботи було розроблено та створено веб-сайт «Національна кухня України». Впровадження даного сайту є логічним та ефективним кроком для залучення нових туристів та представлення інформації про регіональні страви, типові інгредієнти та унікальні кулінарні традиції.

Перед початком проектування та розробки веб-сайту, виконується оцінка його економічної ефективності, що дозволяє прогнозувати потенційні результати та обґрунтувати доцільність впровадження даного сайту у мережі. Розрахунок економічної ефективності є необхідною передумовою для прийняття рішення про розробку та впровадження веб-сайту. Передбачення потенційних результатів дозволяє оцінити наскільки вигідним буде використання сайту та які можливості він надаватиме. Основними економічними показниками є собівартість розробки та ціна, що враховуються при розрахунку потенційного прибутку.

Розрахунок собівартості включає в себе оцінку витрат на проектування та розробку веб-сайту, такі як оплата праці фахівців, витрати на програмне забезпечення та обладнання. Після визначення собівартості, проводиться аналіз ринкової ситуації та встановлюється оптимальна ціна на продукт. Ціна повинна бути конкурентоспроможною і забезпечувати високий рівень попиту на послуги, що надаються через веб-сайт.

Таким чином, розрахунок економічної ефективності перед проектуванням і розробкою веб-сайту дозволяє прийняти обґрунтовані рішення щодо розробки та створення веб-сайту, а також прогнозувати його потенційний прибуток та вплив на фінансові показники компанії.

Давайте розглянемо переваги запропонованого проекту веб-сайту.

Один з головних плюсів цього сайту полягає в його максимально простій структурі та невеликій кількості різних сторінок, що дозволяє його

розробку з використанням мінімальної кількості інструментальних засобів. Це не тільки дозволяє зекономити час на навчання персоналу, але й зменшує витрати, пов'язані з придбанням складних програмних продуктів або іншого спеціалізованого обладнання.

Крім того, важливим перевагою є те, що сайт має статичний характер з погляду наповнення. Це означає, що його вміст не потребує постійних змін і оновлень, що дозволяє уникнути додаткових затрат часу та фінансових ресурсів на підтримку та регулярне оновлення. Такий підхід звільняє компанію від необхідності постійно оновлювати і додавати новий контент, зосереджуючись в основному на сталій та незмінній інформації про діяльність компанії.

Однією з ключових переваг даного сайту є його невеликий обсяг займаного дискового простору. Це означає, що веб-сайт вимагає менше ресурсів для зберігання та обробки даних, що сприяє зменшенню витрат на хостингові послуги. При переході на платний хостинговий майданчик, компанія зможе обрати тарифний план з меншим обсягом дискового простору, що дозволить значно заощадити кошти. Зменшення обсягу займаного дискового простору також має позитивний вплив на продуктивність та швидкодію сайту. Менше обсяг даних, що потребує передач, сприяє прискоренню завантаження сторінок та поліпшує загальний користувацький досвід.

Крім того, невеликий обсяг займаного дискового простору також полегшує резервне копіювання та зберігання даних. Зменшення обсягу даних спрощує процес регулярного створення резервних копій, що є важливим аспектом забезпечення безпеки даних та відновлення роботи сайту у разі потреби. Використання веб-сайту з обмеженим обсягом займаного дискового простору виявляється вигідним з економічної та технічної точок зору.

Таким чином, застосування простої структури та статичного характеру сайту виявляється вигідним з точки зору ефективного використання ресурсів, зокрема часу та коштів, що дозволить зосередити зусилля на інших важливих аспектах діяльності компанії.

Давайте розглянемо конкурентне середовище в контексті тематики запропонованого проекту веб-сайту.

Аналіз конкурентного середовища відіграє важливу роль у плануванні та впровадженні веб-сайту з метою досягнення конкурентних переваг. При розгляді конкурентного середовища варто визначити, які компанії вже присутні на ринку зі схожою тематикою, які послуги або продукти вони пропонують та яким чином вони позиціонуються перед цільовою аудиторією.

Перевагою розробленого веб-сайту про національну кухню України є унікальність та інноваційність його пропозицій, що дозволяють вирізнятися серед конкурентів, тобто усіх інших сайтів з рецептами. Використання інтерактивної мапи України є головним елементом сайту, призначенням якого, є залучити увагу та зацікавленість цільової аудиторії.

Крім того, варто дослідити конкурентні веб-сайти з погляду їхнього дизайну, навігації, якості контенту та взаємодії з користувачами. Це допоможе виявити сильні та слабкі сторони конкурентів і зосередитися на розробці веб-сайту, який буде надавати вищу якість та зручність використання.

Основні переваги веб-сайту «Національна кухня України» у порівнянні з конкурентами:

- свіжий та сучасний дизайн сайту;
- унікальна ідея та її реалізація;
- якісний контент та зручна навігація на сайті.

Розглянемо послідовність процесу проектування веб-сайту. Зазвичай розробка інформаційного веб-сайту включає наступні етапи:

а) початковий етап:

- 1) формулювання основних вимог до сайту та визначення його цілей;
- 2) розробка специфікацій, що описують основні характеристики та

показники сайту;

б) етап зовнішнього проектування:

- 1) розробка архітектури та структури сайту;

- 2) визначення алгоритму побудови сайту;
- 3) виявлення підсистем та окремих складових модулів;
- 4) розробка інтерфейсу;
- в) етап кодування: розробка окремих модулів сайту з використанням HTML, CSS, JavaScript;
- г) етап тестування:
 - 1) налагодження та тестування окремих програмних модулів;
 - 2) комплексне налагодження всього веб-сайту в цілому;
- д) фінальний етап: остаточна корекція веб-сайту на основі здійснених тестувань.

Виконання цих етапів у правильній послідовності допоможе забезпечити успішне проектування веб-сайту про національну кухню України.

Виконаємо розрахунок вартості виготовлення та ціни розробки веб-сайту. У вартість розробки веб-сайту включаються наступні складові витрати:

- основна заробітна плата: витрати, пов'язані з оплатою праці розробників, програмістів та інших спеціалістів, які беруть участь у процесі розробки сайту;
- додаткова заробітна плата: включає додаткові виплати, такі як премії або бонуси, що можуть бути надані команді розробників за досягнення певних цілей або успіхів проекту;
- єдиний соціальний внесок: обов'язковий платіж, що сплачується роботодавцем в державний бюджет на соціальне страхування працівників;
- інші витрати: включають в себе різноманітні додаткові витрати, такі як оплата хостингу, придбання необхідного програмного забезпечення або інших ресурсів, необхідних для розробки та функціонування сайту.

Виконання розрахунку собівартості та встановлення ціни за розробку веб-сайту допоможе визначити вартість проекту та встановити адекватну ціну за надані послуги.

Розробка веб-сайту виконується командою, яка складається зі спеціалістів з різних областей, щоб забезпечити успішну реалізацію проекту.

Основними учасниками команди є дизайнер, програміст-верстальник, а також фронтенд-інженер, кожен з яких відповідає за свої завдання та працює в своєму сегменті процесу розробки.

Дизайнер відповідає за створення естетичного та привабливого зовнішнього вигляду веб-сайту. Він працює над вибором кольорів, шрифтів, композицією, графічними елементами та іншими аспектами дизайну. Дизайнер прагне створити візуальний образ сайту, що привертає увагу, легко сприймається та відображає його концепцію і цілі

Фронтенд-інженер відповідає за розробку клієнтської частини веб-сайту. Він займається програмуванням інтерактивних елементів, реалізацією функціональності та взаємодії з користувачем. Його завдання полягає в створенні зручного та ефективного інтерфейсу, що забезпечує позитивний користувацький досвід.

Програміст-верстальник відповідає за перетворення дизайну та функціональності веб-сайту на кодову основу. Він розробляє HTML та CSS-стилізацію, що визначають вигляд та оформлення сайту. Програміст-верстальник враховує дизайнерські вимоги та співпрацює з фронтенд-інженером для належного впровадження функціональності сбоку клієнта. Він також забезпечує оптимізацію коду, щоб сайт працював швидко та ефективно.

Команда, яка складається з фронтенд-інженера, дизайнера та програміста-верстальника, співпрацює тісно протягом усього процесу розробки веб-сайту – обмінюються ідеями, консультуються між собою та допомагають одне одному для досягнення кращих результатів. Взаємодія між цими спеціалістами дозволяє створити функціональний, естетичний та привабливий веб-сайт, що задовольняє потреби користувачів.

Кожен з членів команди має свої власні навички та експертизу, що використовуються ними для особистого внеску у розробку веб-сайту. Їхні спільне зусилля спрямовані на досягнення високої якості продукту, що відповідає вимогам та очікуванням клієнта.

Робота команди фронтенд-інженера, дизайнера та програміста-верстальника є важливою складовою процесу розробки веб-сайту. Їхні знання, навички та співпраця сприяють створенню функціонального, естетичного та успішного веб-проекту, який задовольняє потреби та очікування користувачів.

Зарплата фронтенд-інженера становить 180,00 грн/год, верстальника – 150 грн/год, дизайнера – 145 грн/год. Тривалість робочого дня кожного з них становить 8 годин. Розробка сайту займає 10 днів.

Додаткова заробітна плата представляє собою винагороду, що виплачується за виконання роботи, що перевищує встановлені норми, за досягнення трудових успіхів та проявлення винахідливості, а також за роботу в особливих умовах. Вона включає різноманітні доплати, надбавки, гарантійні та компенсаційні виплати, що передбачені чинним законодавством. Також до додаткової заробітної плати входять премії, що пов'язані з успішним виконанням виробничих завдань та функцій.

Ці додаткові виплати відображають визнання працівників за їхні досягнення та внесок у роботу. Вони можуть бути виплачені як за результативність у виконанні робочих завдань, так і за особливі умови праці, які вимагають додаткового зусилля та витрат. Додаткова заробітна плата відіграє важливу роль у мотивації працівників до більш ефективної праці та досягнення високих результатів. Вона стимулює працівників до прояву творчості, ініціативи та постійного самовдосконалення.

Додаткова заробітна плата становить 25 % від основної:

$$12640,00 * 0,25 = 3160,00 \text{ грн.}$$

Результати розрахунків наведено у таблиці 6.1.

Єдиний соціальний внесок (ЄСВ) в Україні є обов'язковим податком, який сплачується до державного бюджету підприємствами з метою забезпечення соціального захисту населення.

Таблиця 6.1 – Розрахунок витрат на заробітну плату

Етап	Вид робіт	Виконавець		Годинна ставка, грн	Тривалість виконання, дні	Заробітна плата, грн
		Кількість	Посада			
1. Початковий	Формулювання основних вимог	1	інженер	180,00	2	2880,00
	Розробка специфікацій					
2. Графічна частина	Розробка макету дизайн	1	дизайнер	145,00	2	2320,00
3. Кодування	Верстка	1	програміст	150,00	3	3600,00
4. Тестування	Тестування компонентів	1	програміст	150,00	1	1200,00
	Комплексне налагодження веб-сайту	1	програміст	150,00	1	1200,00
5. Фінальний	Корекція веб-сайту	1	інженер	180,00	1	1440,00
Разом					10	12640,00
Додаткова заробітна плата (25 %)						3160,00
Усього						15800,00

Цей внесок включає в себе обов'язкові страхові внески на соціальне страхування, зокрема на загальнообов'язкове державне соціальне страхування та обов'язкове страхування від нещасних випадків на виробництві та професійних захворювань. Зібрані кошти від ЄСВ направляються на фінансування різних соціальних програм та послуг, зокрема пенсійної системи, медичного страхування, соціального забезпечення, допомоги по безробіттю та інших соціальних заходів. Єдиний соціальний внесок є важливим джерелом фінансування соціальних потреб держави і відіграє важливу роль у забезпеченні соціальної захищеності громадян та стабільності соціально-економічної системи країни.

Ставка єдиного соціального внеску становить 22 % від величини основної і додаткової заробітної плати:

$$15800,00 * 0,22 = 3476,00 \text{ грн.}$$

До інших витрат слід віднести витрати на електроенергію і плату на обслуговування електронно-обчислювальних машин (ЕОМ).

Витрати на електроенергію розраховуються на основі споживаної потужності пристроїв та тарифу на електроенергію. У даному випадку, для оцінки витрат, передбачається використання двох комп'ютерів з потужністю 0,8 кВт/год. Вартість одиниці електроенергії, тобто одного кіловат-години (кВт/год), становить 1,68 грн.

При розрахунку витрат на електроенергію враховується споживана потужність пристроїв, що в даному випадку складає 0,8 кВт/год. Це означає, що за кожну годину роботи двох комп'ютерів витрачається 0,8 кіловат електроенергії.

Тариф на електроенергію встановлений у розмірі 1,68 грн за кВт/год. Це означає, що за кожен кіловат-годину спожитої електроенергії необхідно сплатити 1,68 грн. Час використання електроенергії в процесі розробки:

$$8 * 10 = 80 \text{ год.}$$

Витрати на електроенергію визначаються за формулою:

$$K_e = P_n \times C_e \times T, \quad (6.1)$$

де P_n – потужність споживана, кВт/год.;

C_e – вартість 1 кВт/год.;

T – кількість годин роботи обладнання.

Плата за електроенергію складе:

$$0,8 * 1,68 * 80 * 2 = 215,04 \text{ грн.}$$

Витрати на обслуговування ЕОМ визначаються на основі їх вартості та тривалості експлуатації. Після закінчення тривалості експлуатації, що

зазвичай не перевищує 3 роки, ЕОМ підлягає заміні. Протягом року ЕОМ використовуються 254 робочих дні. Використовуючи (6.1) можемо отримати:

$$(15000,00 / (3 * 8 * 254)) * 80 = 196,85 \text{ грн.}$$

У зв'язку з тим, що проект буде впроваджуватися лише для однієї компанії, сума витрат на розробку складає 19687,89 грн:

$$(15800,00 + 3476,00 + 215,04 + 196,85)/1=19687,89 \text{ грн.}$$

Здійснимо обчислення величини прибутку, отриманого від реалізації проекту розробки, з урахуванням рентабельності на рівні 30 %:

$$19687,89 * 0,3 = 5906,38 \text{ грн.}$$

Розрахуємо ціну розробки сайту без податку на додану вартість (ПДВ):

$$19687,89 + 5906,38 = 25594,27 \text{ грн.}$$

Здійснимо обчислення суми ПДВ, що становить 20 % від ціни без урахування ПДВ.

$$25594,27 * 0,2 = 5118,85 \text{ грн.}$$

Проаналізуємо вартість веб-сайту з урахуванням податку на додану вартість (ПДВ) і здійснимо розрахунок кінцевої ціни.

$$25594,27 + 5118,85 = 30713,12 \text{ грн.}$$

Результати розрахунків наведено у таблиці 6.2.

Таблиця 6.2 – Розрахунок витрат на розробку та ціни веб-сайту

Стаття витрат	Сума, грн
Основна заробітна плата	12640,00
Додаткова заробітна плата	3160,00
Єдиний соціальний внесок	3476,00
Витрати на обслуговування ЕОМ	196,85
Витрати на електроенергію	215,04
Собівартість розробки сайту	19687,89
Прибуток	5906,38
Ціна без ПДВ	25594,27
Податок на додану вартість (ПДВ)	5118,85
Ціна з урахуванням ПДВ	30713,12

Отже, на основі проведених розрахунків повна вартість розробки сайту становить 30713,12 грн. Термін виконання всіх етапів проекту складає 10 днів з участю одного програміста, дизайнера і фронтенд-інженера. Очікувана сума прибутку оцінюється на рівні 5906,38 грн.

ВИСНОВКИ

В даній кваліфікаційній роботі був розроблений веб-сайт «Національна кухня України», який надає широку інформацію про національну кухню України. Сайт створений з метою представлення традицій, особливостей регіональних кулінарних стилів, популярних інгредієнтів, спецій та страв української кухні. В результаті відвідувачі сайту мають змогу ознайомитися з типовими методами приготування та популярними рецептами.

У розробці сайту враховані сучасні тенденції веб-розробки, забезпечуючи приємний дизайн та інтуїтивно зрозумілу навігацію. Крім того, сайт розроблений з урахуванням адаптивного дизайну, щоб забезпечити його належне відображення на різних пристроях, таких як комп'ютери, планшети та смартфони.

Також успішно реалізовані певні завдання. Насамперед проведено огляд існуючих інструментів та інформаційних технологій, що використовуються для створення веб-сайтів. Потім визначено структуру та дизайн самого веб-сайту, які забезпечують його ефективне функціонування. Основний функціонал веб-сайту успішно реалізований, це дозволяє користувачам взаємодіяти з додатком та використовувати його можливості. Нарешті, проведено функціональне тестування веб-сайту, щоб переконатися в його коректній роботі та відповідності заданим вимогам.

Результатом роботи став функціональний веб-сайт «Національна кухня України», що успішно відповідає поставленим цілям та вимогам. Сайт надає цінну інформацію про українську кухню, сприяє поширенню культурних знань та заохочує зацікавленість відвідувачів у вивченні та експериментах з українськими стравами.

Перед початком розробки виконано оцінку економічної ефективності сайту. Це дозволило прогнозувати потенційні прибутки та собівартість розробки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Гончаренко Д. Як створити сайт з нуля. URL: <https://hostpro.ua/blog/ua/how-to-build-a-website> (дата звернення: 15.05.2023).
2. Documenting web technologies, including CSS, HTML, and JavaScript. URL: <https://developer.mozilla.org/ru/> (Lastaccessed: 15.05.2023).
3. 15 найкращих безкоштовних редакторів коду для ідеального кодування. URL: <https://uk.myservername.com/top-15-best-free-code-editors> (дата звернення: 15.05.2023).
4. Спіридонова Х. Розробка веб-додатків: основні етапи. URL: <https://www.purrweb.com/blog/what-is-a-web-app/> (дата звернення: 15.05.2023).
5. Етапи створення веб-сайтів. URL: <https://webtune.com.ua/statti/web-rozrobka/etapy-stvorennya-veb-sajtiv/> (дата звернення: 15.05.2023).
6. Адаптивний дизайн сайту: в чому переваги і чому це важливо. URL: <https://ag.marketing/blog/adaptyvnyy-dyzayn-saytu/> (дата звернення: 15.05.2023).
7. Веб-розробка, з чого розпочати навчання. URL: <https://kiev.itstep.org/blog/web-development-where-to-start-learning> (дата звернення: 15.05.2023).
8. Морган Н.Г. JavaScript для дітей: навч. посіб. Львів: Видавництво Старого Лева, 2021. 408 с.
9. Фрімен Е. Head First. Вивчаємо HTML, XHTML та CSS: навч. посіб. Київ: Фабула, 2019. 720 с.
10. Фрімен Е., Робсон Е. Head First. Програмування на JavaScript: навч. посіб. Київ: Фабула, 2022. 672 с.
11. Шарко М.В., Мешкова-Кравченко Н.В., Радкевич О.М. Економіка підприємства: навч. посіб. Ч. 1. Херсон, 2014. 434 с.
12. Вимоги до створення сайту. URL: <https://recommerce.com.ua/vimogi-do-stvorennya-saitu> (дата звернення: 15.05.2023).
13. Ключові етапи створення сайту. URL: <https://pbb.lviv.ua/en/blog/about-web-development/the-main-stages-of-creating-a-website/> (дата звернення: 15.05.2023).

14. Методичні вказівки з виконання кваліфікаційної роботи для студентів денної та заочної форми навчання першого (бакалаврського) рівня вищої освіти спеціальності 186 «Видавництво та поліграфія» за освітньою програмою «Видавничо-поліграфічна справа» / В.П. Ткаченко, А.В. Бізюк, О.В. Вовк, І.М. Єгорова, В.Ф. Челомбійко. Харків: ХНУРЕ, 2020. 68 с.

15. Посібник з JavaScript. URL: <https://metanit.com/web/javascript/> (дата звернення: 15.05.2023).

16. Як розробити дизайн сайту: просто про складне. URL: <https://itspace.company/blog/iak-rozrobyty-dyzaun-saytu-prosto-pro-skladne> (дата звернення: 15.05.2023).