

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

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти бакалавр

на тему: **«Розроблення шаблону адаптивного вебсайту для молодіжного
кафе у формі лендінгу»**

Виконав: здобувач вищої освіти
за освітньою програмою
Інформаційні управляючі системи
спеціальності 126 Інформаційні
системи та технології
ступеня вищої освіти бакалавр
групи 126ІСТ_бд_2022
Бабінський Богдан Юрійович
Керівник: Уткін Юрій Вікторович
Рецензент: Яхін Сергій Валерійович

Полтава – 2026 року

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

Освітня програма Інформаційні управляючі системи
Спеціальність 126 Інформаційні системи та технології
Рівень вищої освіти перший (бакалаврський)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ **Юрій УТКІН**
«10» червня 2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
Бабінського Богдана Юрійовича

1. Тема кваліфікаційної роботи:
«Розроблення шаблону адаптивного вебсайту для молодіжного кафе у формі лендінгу»,
Керівник роботи: к. т. н., доцент, доцент кафедри інформаційних систем та технологій Уткін Юрій Вікторович
Затверджено наказом закладу вищої освіти від «10» квітня 2026 року № 383-ст
2. Строк подання здобувачем вищої освіти роботи «08» червня 2026 року.
3. Вихідні дані до роботи: статистичні, науково-популярні джерела мережі інтернет, специфікації з веброзробки <https://www.w3.org/Translations/WCAG21-ua/>, бібліотеки jQuery <https://jquery.com/download/>, статистичні дані аналітичних компаній <https://www.internetlivestats.com/>, <https://w3techs.com/>, дані офіційних сайтів <https://thehost.ua/>, <https://cases.media/> та ін.
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):
Розділ 1 Теоретичні аспекти базових технологій проектування вебсайтів
Розділ 2. Проектування шаблону адаптивного вебсайту молодіжного кафе
Розділ 3 Програмна реалізація процесу розроблення промо-сайту для спеціалізованої кав'ярні
5. Перелік графічного матеріалу: схеми, рисунки, діаграми за темою та об'єктом дослідження

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання отримав
Оцінювання економічної ефективності результатів дослідження	Дивнич О. Д., к. е. н., доцент, доцент кафедри економіки та публічного управління	01.04.2026 р.	29.05.2026 р.

7. Дата видачі завдання «10» червня 2025 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Вибір і затвердження теми роботи	02.06.2025 р. – 04.06.2025	
2	Складання і затвердження розгорнутого плану та завдання на кваліфікаційну роботу	05.06.2025 р. – 10.06.2025 р.	
3	Опрацювання джерел інформації	11.06.2025 р. – 15.06.2025 р.	
4	Збір, вивчення і обробка інформації, необхідної для виконання роботи	16.06.2025 р. – 20.06.2025 р.	
5	Виконання теоретико-методологічного розділу роботи	08.09.2025 р. – 01.11.2025 р.	
6	Виконання дослідницько-аналітичного розділу роботи	19.01.2026 р. – 14.02.2026 р.	
7	Виконання проєктно-рекомендаційного розділу роботи	16.02.2026 р. – 31.03.2026 р.	
8	Оцінювання економічної ефективності результатів дослідження	01.04.2026 р. – 29.05.2026 р.	
9	Оформлення тексту роботи	01.06.2026 р. – 06.06.2026р.	
10	Попередній захист роботи на кафедрі	08.06.2026 р.	
11	Доопрацювання роботи з урахуванням зауважень і пропозицій	09.06.2026 р. – 10.06.2026 р.	
12	Нормоконтроль	11.06.2026 р. – 15.06.2026 р.	
13	Захист кваліфікаційної роботи	16.06.2026 р. - 18.06.2026 р.	

Здобувач вищої освіти

Богдан БАБІНСЬКИЙ

Керівник роботи

Юрій УТКІН

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1 ТЕОРЕТИЧНІ АСПЕКТИ БАЗОВИХ ТЕХНОЛОГІЙ ПРОЄКТУВАННЯ ВЕБСАЙТІВ	9
1.1 Теорія розвитку вебсайтів та особливості використання у сфері бізнесу	9
1.2 Характеристика сучасних базових технологій розроблення вебсайтів	15
1.3 Характеристики дизайну вебсайтів у контексті сучасних трендів	23
РОЗДІЛ 2 ПРОЄКТУВАННЯ ШАБЛОНУ АДАПТИВНОГО ВЕБСАЙТУ МОЛОДІЖНОГО КАФЕ	27
2.1 Побудова концепції розроблення шаблону сайту для втілення бізнес- плану молодіжного кафе	27
2.2 Обґрунтування вибору методів та середовища створення сайту	31
2.3 Проєктування макету шаблону сайту з використанням Figma	36
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЦЕСУ РОЗРОБЛЕННЯ ПРОМО- САЙТУ ДЛЯ СПЕЦІАЛІЗОВАНОЇ КАВ'ЯРНІ	41
3.1 Розроблення основного коду промо-сайту і забезпечення функцій	41
3.2 Удосконалення результату розробки шаблону сайту для адаптивності на екранах смартфонів та елементи SEO-оптимізації	49
3.3 Економічний аналіз варіантів розробки вебсайту	53
ВИСНОВКИ	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	59
ДОДАТКИ	63

ВСТУП

В сучасному цифровому світі вебсайт є одним із ключових інструментів представлення закладів громадського харчування в мережі інтернет. Особливо важливим є створення ефективних вебресурсів для молодіжних кафе, основною цільовою аудиторією яких є користувачі мобільних пристроїв та соціальних мереж. Більшість потенційних відвідувачів здійснює пошук інформації про заклад, перегляд меню, ознайомлення з акціями та бронювання столиків за допомогою смартфонів, що обумовлює необхідність використання адаптивного дизайну.

Актуальність теми кваліфікаційної роботи зумовлена необхідністю створення сучасних адаптивних вебресурсів для закладів громадського харчування, орієнтованих на мобільних користувачів, а також потребою розроблення універсальних шаблонів вебсайтів, які можуть використовуватися як основа для швидкого створення ефективних лендінгових сторінок різних кафе та інших закладів сфери обслуговування. Такий підхід сприяє стандартизації процесу веброзробки, зниженню вартості впровадження вебрішень та підвищенню доступності сучасних цифрових інструментів для малого бізнесу.

Метою кваліфікаційної роботи є проєктування та розроблення шаблону адаптивного вебсайту молодіжного кафе у формі лендінгу, який забезпечує зручне представлення інформації про заклад на різних типах пристроїв та сприяє підвищенню ефективності взаємодії з цільовою аудиторією.

Завдання кваліфікаційної роботи:

- провести аналіз сучасних тенденцій розроблення вебсайтів для закладів громадського харчування;
- дослідити особливості структури та функціонування лендінгових вебсайтів;
- проаналізувати принципи адаптивного вебдизайну та технології його реалізації;

- визначити вимоги до дизайну та функціональності вебсайту молодіжного кафе;
- розробити прототип інтерфейсу майбутнього вебсайту;
- створити дизайн-макет адаптивного лендінгу;
- реалізувати шаблон вебсайту засобами сучасних вебтехнологій;
- провести тестування коректності роботи та адаптивності вебсайту на різних пристроях;
- оцінити ефективність розробленого рішення та можливості його практичного використання.

Об'єктом дослідження кваліфікаційної роботи є процес проєктування та розроблення вебсайтів для закладів громадського харчування.

Предметом дослідження є методи, засоби та технології створення адаптивного лендінгового вебсайту для молодіжного кафе.

Методи дослідження: інформаційно-пошуковий, аналітико-синтетичний, порівняльний, абстрактний, проєктування інтерфейсів користувача, адаптивної верстки, методи оцінювання ефективності вебресурсу та інші.

Інформаційну базу кваліфікаційної роботи складають: наукові статті, тематична література, специфікації W3C стосовно технологій HTML CSS JavaScript, маркетингові дослідження функцій вебсайтів для закладів харчування, статистичні дані з аналітичних звітів міжнародних компаній і груп, які розміщені в мережі інтернет у вільному доступі, нормативні документи в галузі ІТ тощо.

Практична значущість роботи: Практична значимість роботи полягає в розробленні готового шаблону адаптивного вебсайту у формі лендінгу для молодіжного кафе, який може бути використаний як основа для створення вебресурсів закладів громадського харчування без необхідності повного проєктування з нуля. Розроблений шаблон забезпечує коректне відображення контенту на персональних комп'ютерах, планшетах і мобільних пристроях, має сучасний інтерфейс та зручну структуру подання інформації.

Апробація результатів роботи відбувалася шляхом оприлюднення доповіді на XXI щорічній студентській науковій конференції «Сучасні інформаційні технології та інноваційні методики в економіці, менеджменті та бізнесі» 22 квітня 2026 року.

Структура і обсяг кваліфікаційної роботи: пояснювальна записка має 58 сторінок і складається зі вступу, трьох розділів; висновків, списку використаних джерел, додатків. Робота містить 38 рисунків, 7 таблиць, посилання на 44 інформаційних джерела.

РОЗДІЛ 1

ТЕОРЕТИЧНІ АСПЕКТИ БАЗОВИХ ТЕХНОЛОГІЙ ПРОЄКТУВАННЯ ВЕБСАЙТІВ

1.1 Теорія розвитку вебсайтів та особливості використання у сфері бізнесу

Перший у світі доступний вебсайт з'явився 20 грудня 1990 року. На ньому розміщувався опис нової технології World Wide Web (WWW), заснованої на протоколі передачі даних HTTP, системі адресації URL та мові гіпертекстової розмітки HTML [1].

Відомим є факт, що мова гіпертекстової розмітки HTML (HyperText Markup Language) була розроблена британським вченим Тімом Бернерсом-Лі у 1986-1991 рр. у стінах Центру ядерних досліджень (CERN) у Женеві в Швейцарії і від початку створювалася як мова для обміну технічною та науковою документацією, яку б могли використовувати люди, які не є фахівцями в області верстання [2].

Вебсайт – це набір файлів і документів, які поєднані спеціальними зв'язками – гіперпосиланнями, знаходяться на якомусь сервері і об'єднані під одне доменне ім'я. Вебсайт для бізнесу виконує такі функції: реклама послуг, ідей, продукції; продаж товарів, послуг, ідей; безкоштовне надання інформації; підтримка клієнтів та багато інших [3].

Винайдення технологій розробки сайтів по суті стало революційною подією. Стандартами та специфікаціями ключових технологій розроблення вебсайтів займається W3C-консорціум, який від початку і до сьогодні очолює легендарний британець Тім Бернес-Лі. Протягом тридцяти з лишнім років мова гіпертексту HTML розвивалася і удосконалювалася завдяки відкритості платформи W3C [4]. Всі специфікації, пояснення, приклади розробники можуть знайти на офіційному вебсайті консорціуму. Стандарти W3C визначають відкриту вебплатформу для розробки додатків, яка має

безпрецедентний потенціал, щоб дозволити розробникам створювати насичений інтерактивний досвід на основі величезних сховищ даних, доступних на будь-якому пристрої [5]. Хоча межі платформи продовжують розвиватися, лідери галузі говорять майже в унісон про те, що HTML5 стане наріжним каменем для цієї платформи. Але повна потужність платформи спирається на багато інших технологій, які створюють W3C та її партнери, включаючи CSS, SVG, WOFF, стек семантичного вебу, XML та різноманітні API. W3C розробляє ці технічні специфікації та вказівки за допомогою процесу, розробленого для досягнення максимального консенсусу щодо змісту технічного звіту, забезпечення високої технічної та редакційної якості та отримання схвалення W3C та ширшої спільноти.

Вебсайт складається із пов'язаних між собою вебсторінок. Вебсторінка представляє собою текстовий файл із розширенням *.html, який містить текст, який буде зображено на сайті, а також HTML теги, які й визначають, як буде розміщуватись інформація на сторінці. HTML документи зручні тим, що для їх перегляду підходить майже будь-який браузер [6].

Доменне ім'я – це унікальна адреса сайту, яка використовується для його специфікації та пошуку в мережі. Воно складається з літер та цифр, а також може містити символ «-». Мінімальна довжина – 1 символ, максимальна – 63. Наприклад, «www.pdau.edu.ua» є доменним ім'ям Полтавського державного аграрного університету, «www.statista.com» - комерційний домен.

Домен першого рівня – це права частина домену після крапки. Цю частину може ставити лише ICANN (Інтернет корпорація з присвоєння імен та номерів), яка діє в інтересах всього інтернет-товариства з дотриманням міжнародних конвенцій та місцевих законодавств. Домен може відрізнятись за належністю до територіальних зон, або типів організації, наприклад: .ua, .us, .fr, .com, .edu, .net. Домен другого рівня – це унікальна назва сайту. Наприклад, statista.com. Домен третього рівня зазвичай використовується для створення невеликих сайтів, або підрозділів головного сайту, наприклад, форум. Як приклад домену третього рівня може бути test.site.ua [7].

Аналіз архітектури сучасного інформаційного простору інтернету станом на лютий 2026 р. вказує на значний дисбаланс між номінальною кількістю зареєстрованих вебресурсів та обсягом фактично функціонуючих сайтів. За даними глобального моніторингу компанії Netcraft (Netcraft Web Server Survey [8]), загальна кількість вебсайтів досягла позначки у 1,42 млрд одиниць. Проте, детальне вивчення структури мережеских вузлів виявляє, що лише 15,0% (близько 213 млн) є активними вебресурсами, які генерують унікальний контент та обслуговують користувачів [9]. Переважна більшість інформаційного масиву мережі (85,0%, або понад 1,20 млрд сайтів) класифікується як "інертний кіберпростір". До цієї категорії належать:

- парковані домени (Parked Domains): викуплені кіберсквотерами або інвесторами з метою перепродажу;
- службові та технічні сторінки: тимчасові заглушки хостинг-провайдерів та реєстраторів;
- закинуті (Dormant) проєкти: вебресурси, підтримку яких припинено власниками, але термін делегування доменного імені яких ще не вичерпався.

Діаграма розподілу активних/неактивних вебсайтів станом на лютий 2026 р. За даними [8] показано на рис. 1.1.

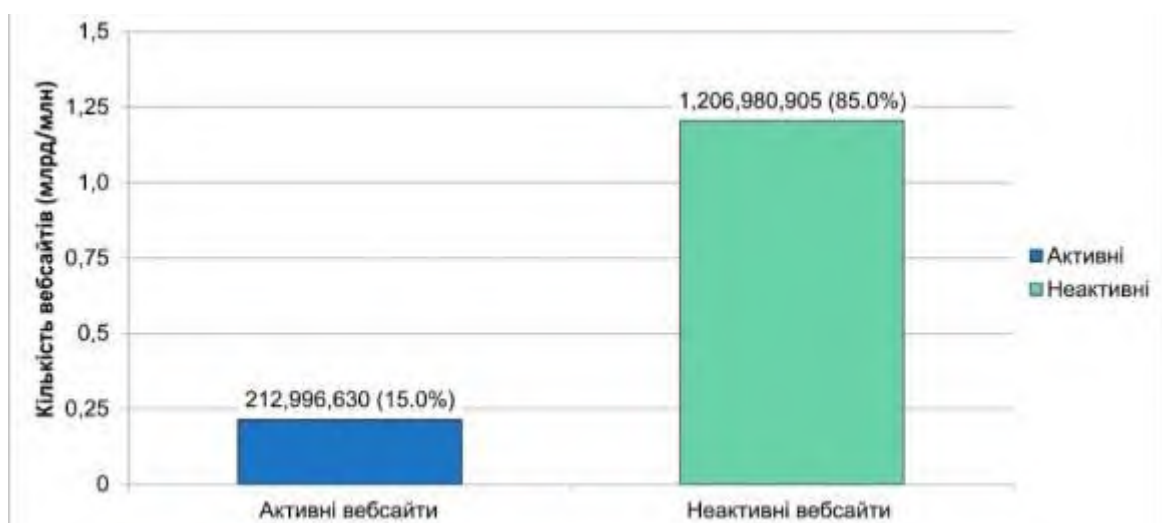


Рисунок 1.1 – Порівняння кількості активних і неактивних вебсайтів мережі інтернет станом на лютий 2026 р. (побудовано за даними [8])

Таке співвідношення (див. рис. 1.1) свідчить про високий рівень комерціалізації доменного ринку, де значна частина інфраструктури використовується як цифровий актив або спекулятивний інструмент, а не як безпосередня платформа для комунікації чи надання послуг. Це створює додаткове навантаження на систему доменних імен (DNS) та вимагає оптимізації алгоритмів пошукових систем для фільтрації неактивного контенту. Більшість відомих і популярних світових вебдодатків, мереж була розроблена вже до 2010 р. (табл. 1.1). Необхідність професійно створеного сайту усвідомила велика кількість керівників бізнесу. Більшість фірм вкладають кошти для надання сайту сучасних функцій і ознак: це необхідний чинник для, який дозволяє привертати додаткову аудиторію [8].

Таблиця 1.1 – Хронологія появи відомих світових вебзастосунків [10]

Рік	Кількість вебсайтів	Зміна за рік	Кількість користувачів інтернету	Користувачів на сайт, осіб	Назви вебдодатків, платформ, мереж, що з'явилися
2010	206 956 723	-13%	2 045 865 660	9,9	Pinterest, Instagram
2009	238 027 855	38%	1 766 206 240	7,4	
2008	172 338 726	41%	1 571 601 630	9,1	Dropbox
2007	121 892 559	43%	1 373 327 790	11,3	Tumblr
2006	85 507 314	32%	1 160 335 280	13,6	Twitter
2005	64 780 617	26%	1 027 580 990	16	YouTube, Reddit
2004	51 611 646	26%	910 060 180	18	The facebook, Flickr
2003	40 912 332	6%	778 555 680	19	WordPress, LinkedIn
2002	38 760 373	32%	662 663 600	17	
2001	29 254 370	71%	500 609 240	17	Wikipedia
2000	17 087 182	438%	413 425 190	24	Baidu
1999	3 177 453	32%	280 866 670	88	PayPal
1998	2 410 067	116%	188 023 930	78	Google
1997	1 117 255	334%	120 758 310	108	Yandex, Netflix
1996	257 601	996%	77 433 860	301	
1995	23 500	758%	44 838 900	1,908	Altavista, Amazon, AuctionWeb
1994	2 738	2006%	25 454 590	9,297	Yahoo
1993	130	1200%	14 161 570	108,935	
1991	1				World Wide Web Project

Аналіз розвитку вебтехнологій (див. табл. 1.1) свідчить про те, що вебсайти пройшли тривалий шлях еволюції від простих статичних сторінок до складних інтерактивних інформаційних систем. Якщо перші вебресурси виконували виключно інформаційну функцію, то сучасні вебсайти стали важливим інструментом ведення бізнесу, маркетингових комунікацій, продажу товарів та послуг, а також взаємодії з клієнтами.

Вебсайт – це ефективний інструмент маркетингу, який захоплює увагу аудиторії. Сайт повинен зацікавити користувача, а потім підштовхнути його до виконання якоїсь дії, наприклад до покупки товару.

З розвитком електронної комерції та цифрових технологій змінилася і роль вебсайтів у діяльності підприємств. Сьогодні вебсайт часто є першим джерелом інформації, з яким взаємодіє потенційний клієнт. Від якості його реалізації залежить не лише формування першого враження про компанію, але й подальше рішення користувача щодо співпраці, придбання товарів або замовлення послуг. Вебсайти можуть виконувати різноманітні функції залежно від специфіки діяльності організації, звідки і різні форми, типи (табл. 1.2).

Таблиця 1.2 – Основні типи вебсайтів та їх характеристика

Тип вебсайту	Основне призначення	Особливості
Корпоративний сайт	Представлення компанії в мережі Інтернет	Інформація про діяльність, контакти, послуги
Інтернет-магазин	Продаж товарів онлайн	Каталог товарів, кошик, онлайн-оплата
Інформаційний портал	Надання інформації користувачам	Велика кількість контенту та новин
Блог	Публікація авторських матеріалів	Регулярне оновлення контенту
Форум	Комунікація між користувачами	Обговорення тем, створення спільнот
Лендінг	Просування товару, послуги або події	Односторінкова структура та висока конверсія
Вебдодаток	Виконання спеціалізованих функцій	Інтерактивність та складна бізнес-логіка

З наведених типів особливу популярність останніми роками отримали лендінгові сторінки. Вони дозволяють максимально концентрувати увагу користувача на конкретній пропозиції та мінімізують кількість відволікаючих

факторів. Лендінг є одним із найбільш ефективних інструментів інтернет-маркетингу, оскільки орієнтований на досягнення конкретної цільової дії з боку відвідувача [11].

Паралельно з розвитком функціональних можливостей вебресурсів відбувалася зміна підходів до їх проектування. Однією з найважливіших тенденцій сучасної веброзробки стало поширення мобільних пристроїв. За даними численних досліджень цифрового ринку, більше половини світового вебтрафіку формується саме за рахунок смартфонів і планшетів. Це призвело до необхідності створення вебресурсів, здатних коректно відображатися на екранах різних розмірів. У відповідь на ці виклики виникла концепція адаптивного вебдизайну (Responsive Web Design) [12]. Основною ідеєю адаптивного підходу є автоматичне налаштування структури сторінки, розмірів елементів інтерфейсу та розташування контенту залежно від параметрів пристрою користувача. Завдяки цьому забезпечується комфортна взаємодія з вебресурсом незалежно від того, чи використовується персональний комп'ютер, планшет або смартфон [13].

Адаптивність стала одним із ключових критеріїв якості сучасного вебсайту. Вона позитивно впливає на поведінкові фактори користувачів, швидкість завантаження сторінок, доступність інформації та ефективність просування ресурсу в пошукових системах. Пошукові алгоритми приділяють значну увагу мобільній оптимізації вебсайтів, що безпосередньо впливає на їх позиції у результатах пошуку.

Важливим напрямом розвитку вебтехнологій стало вдосконалення користувацького інтерфейсу та досвіду взаємодії користувача із системою. У сучасній практиці широко використовуються концепції UI (User Interface) та UX (User Experience), які спрямовані на забезпечення максимальної зручності використання вебресурсів [14]. Вдалий інтерфейс дозволяє користувачеві швидко знаходити необхідну інформацію, виконувати потрібні дії та отримувати позитивний досвід від взаємодії з сайтом. Основні переваги використання вебсайтів у бізнесі наведено в табл. 1.3.

Таблиця 1.3 – Переваги використання вебсайтів для бізнесу

Перевага	Характеристика
Доступність інформації	Клієнти можуть отримати необхідну інформацію у будь-який час
Розширення аудиторії	Залучення потенційних клієнтів незалежно від місця їх перебування
Маркетингова підтримка	Просування товарів та послуг через цифрові канали
Формування іміджу	Підвищення довіри до компанії та бренду
Зворотний зв'язок	Отримання звернень, відгуків та пропозицій
Автоматизація процесів	Онлайн-бронювання, заявки, консультації
Зменшення витрат	Скорочення витрат на традиційні рекламні канали

Сфера громадського харчування є однією з тих галузей, де вебсайти набули особливого значення. Відвідувачі кафе та ресторанів дедалі частіше знайомляться із закладом через мережу інтернет ще до його відвідування. Користувачі переглядають меню, фотографії інтер'єру, контактну інформацію, години роботи, акційні пропозиції та відгуки інших клієнтів. У результаті вебсайт стає важливим елементом маркетингової стратегії закладу..

1.2 Характеристика сучасних базових технологій розроблення вебсайтів

Розроблення сучасного вебсайту є комплексним процесом, який включає етапи аналізу вимог, проектування структури та інтерфейсу, програмної реалізації, тестування та подальшого впровадження вебресурсу. Якість кінцевого результату значною мірою залежить від правильного вибору технологій розроблення та їх раціонального поєднання. Базовими технологіями створення сучасних вебсайтів є HTML, CSS та JavaScript, які формують основу фронтенд-розробки та забезпечують структурування контенту, його візуальне оформлення та інтерактивну взаємодію з користувачем.

Структура – це логіка побудови сайту, ієрархічна організація взаємозв'язків сторінок. Продумувати структуру сайту потрібно ще до запуску проєкту, оскільки без чіткої логіки ми отримаємо просто набір документів її

простіше всього уявити у графічному вигляді – блок схема або майнд-карта, на якій чітко видно зв'язки, рівні вкладеності та ієрархія сторінок [14].

Технічно структура сайту відображається у його URL. У кожній сторінки є своя унікальна адреса, в якій прописаний шлях до неї від головної сторінки. Якщо структура сайту продумана грамотно, то жодна сторінка не буде окремо від сайту, і до будь-якої з них у користувача буде доступ. Дуже важливо, щоб користувач швидко розібрався, як влаштований сайт і як йому потрапити на необхідну сторінку [15]. Для цього створюється схема внутрішньої структури майбутнього сайту (рис. 1.2).



Рисунок 1.2 –Схема внутрішньої структури сайту

Зовнішня структура вебсайту в більшості випадків складається з таких основних елементів:

- шапка у верхній частині сторінки, де розташовується логотип, клік по якому повертає на головну сторінку;
- бокове меню може бути як справа, так і зліва; на ньому розміщують елементи навігації і допоміжну інформацію;
- основна частина сайту – те, заради чого користувач заходить на сайт. Основна інформація повинна бути читабельною і розміщуватись на першому-другому екрані;
- нижня частина сторінки. Зазвичай, там знаходиться контакти, кнопки соціальних мереж, додаткова інформація [15].

Після формування структури вебресурсу розпочинається етап його технічної реалізації. На цьому етапі використовуються сучасні вебтехнології, що забезпечують створення інтерфейсу користувача, оформлення сторінок та реалізацію інтерактивних функцій. Основу більшості сучасних вебсайтів становить поєднання HTML, CSS та JavaScript.

На мові HTML автори описують структуру сторінок за допомогою розмітки. Елементи мови позначають частини вмісту, такі як параграф `<p>`, список `<ol>`, `<ul>`, `<li>`, таблиця `<table>` тощо. Сучасні вебсайти створюються переважно на основі стандарту HTML5, який підтримується всіма популярними браузерами та забезпечує використання семантичних елементів розмітки. HTML-документ складається з трьох основних частин: Декларація типу документа (англ. Document type declaration, Doctype), на початку документа, в якій визначається тип документа (DTD).

1. Шапка документа (в межах елемента `<head>`), в якій записано загальні технічні відомості або важлива інформація про документ, яка не відтворюється безпосередньо в браузері.

2. Тіло документа (може знаходитися в елементі `<body>`), в якому міститься основна інформація документа.

На практиці при роботі над зовнішнім виглядом сайту використовуються додаткові добре розвинуті за останнє десятиріччя технології, такі як: CSS (Cascading Style Sheets) – мова опису зовнішнього вигляду документу, JS (JavaScript) – мова сценаріїв для придання інтерактивності вебсторінкам. JS використовується майже всіма сучасними сайтами в інтернеті [16].

Разом із графікою та сценаріями HTML і CSS є основою створення вебсторінок і вебдодатків. Актуальною версією CSS на даний момент є CSS3, більш нова CSS4 іще знаходиться у розробці. По суті, таблиця стилів – це спеціальний розділ в головній частині сайту в тегах `<style>...</style>` або окремий файл стилів `style.css`, в якому описується, як буде виглядати кожен із елементів на сайті [16]. В такому разі на цей файл необхідне спеціальне посилання з файлу розмітки `index.html`: `<link rel="stylesheet" href="styles.css">`.

Синтаксис CSS (рис. 1.3) має порівняно простий синтаксис і використовує набір англійських слів для найменування складових стилю [16].

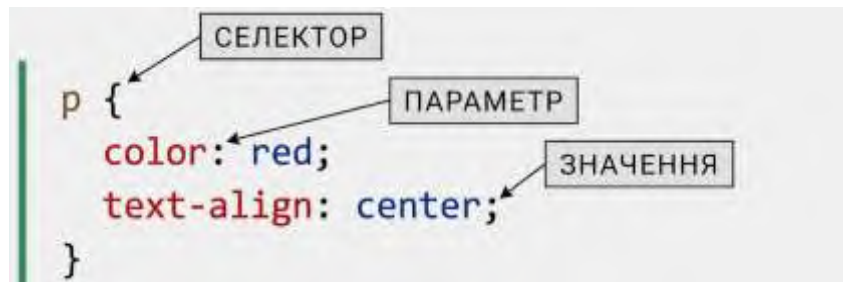


Рисунок 1.3 – Приклад синтаксису CSS та призначення обов’язкових знаків

Стилі складаються зі списку правил. Кожне правило має один або більше селектор (англ. selector) та блок визначення (англ. declaration block). Блок визначення складається з оточеного фігурними дужками списку властивостей.

1. Зовнішні таблиці стилів (англ. stylesheet), найчастіше окремий файл або файли .css.
2. Внутрішні таблиці стилів, включені як частина документу або блоку.
3. Стилі для окремого елемента.
4. Стилі користувача.
5. Локальний .css-файл, вказаний користувачем для використання на сторінках і вказаний в налаштуваннях браузера (наприклад Chrom).
6. Стилі переглядача (браузера).
7. Стандартний стиль переглядача, наприклад стандартні стилі для елементів, визначені браузером, використовуються коли немає інформації про стиль елемента або вона неповна.

Стандарт CSS визначає порядок та діапазон застосування стилів, тобто, в якій послідовності і для яких елементів застосовуються стилі. Таким чином, використовується принцип каскадності, коли для елементів вказується лише та інформація про стилі, що змінилася або не визначена загальнішими стилями.

До переваг використання CSS відносять, зокрема, досягнення єдиного стильового оформлення однакових елементів, економія та раціоналізація коду, швидкість завантаження сторінки та інші. CSS – мова для опису представлення

вебсторінок, включаючи кольори, макет і шрифти. Це дозволяє адаптувати презентацію до різних типів пристроїв, таких як великі екрани, маленькі екрани або принтери. CSS не залежить від HTML і може використовуватися з будь-якою мовою розмітки на основі XML. Відокремлення HTML від CSS забезпечує спільний доступ до таблиць стилів між сторінками та адаптацію сторінок до різних середовищ. Сторінки зменшуються в об'ємі, стають більш структурованими, оскільки інформація про стиль відділена від тексту та має певні правила застосування завдяки тому, що браузері можуть кешувати стилістичні дані кожної сторінки, а не завантажувати їх окремо для кожної.

Існують також CSS-фреймворки, які ще більш спрощують роботу верстальника, серед яких найпоширеніші: Bootstrap, Foundation, Tailwind. А також CSS-препроцесори – надбудови над CSS, які дозволяють писати спрощені синтаксичні конструкції, що підвищує простоту написання і редагування коду. Серед популярних препроцесорів можна виділити: Less, Sass (SCSS), Stylus [17].

Сучасна веброзробка неможлива без використання технологій адаптивного дизайну. Адаптивна верстка (Responsive Web Design) являє собою підхід до проєктування вебсторінок, за якого інтерфейс автоматично пристосовується до різних розмірів екранів і типів пристроїв. Основною метою адаптивного дизайну є забезпечення комфортного перегляду вебресурсу незалежно від того, чи використовує користувач персональний комп'ютер, планшет або смартфон. Для реалізації адаптивності в CSS використовуються медіазапити (Media Queries), які дозволяють задавати різні параметри відображення сторінки залежно від ширини екрана, його орієнтації та інших характеристик пристрою.

Для побудови гнучких макетів широко застосовуються технології Flexbox та CSS Grid. Flexbox призначений для розташування елементів уздовж одного напрямку та спрощує вирівнювання компонентів інтерфейсу. CSS Grid дозволяє створювати складні двовимірні макети сторінок і забезпечує високий рівень контролю над розташуванням блоків (табл. 1.4).

Таблиця 1.4 – Технології адаптивної верстки

Технологія	Призначення	Переваги
Media Queries	Адаптація стилів під різні пристрої	Простота налаштування
Flexbox	Одновимірне розташування елементів	Гнучкість побудови інтерфейсу
CSS Grid	Двовимірне компонування сторінки	Зручність створення складних макетів
Relative Units (% , rem, vw, vh)	Масштабування елементів	Коректне відображення на різних екранах

Мова програмування JavaScript використовується програмістами для створення динамічних інтерактивних вебсторінок. Мова JavaScript настільки популярна, що її використання сягає майже 98% [19]. JavaScript (JS) – динамічна, об’єктно-орієнтована прототипна мова програмування. Найбільш часто використовується для створення сценаріїв вебсторінок, які дозволяють користувачеві взаємодіяти з вебсторінкою, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та вигляд вебсторінки.

JavaScript є скриптовою мовою програмування з динамічною типізацією та є прототипом (підмножиною об’єктно-орієнтованої). Окрім прототипної JavaScript частково підтримує інші парадигми програмування (імперативні та функціональні), а також деякі відповідні архітектурні властивості; це включає динамічну та слабку типізацію, автоматичне керування пам’яттю, прототипне наслідування та функції як об’єкти першого класу. Мова JavaScript використовується для таких задач:

1. Написання сценаріїв вебсторінок для надання їм інтерактивності.
2. Створення односторінкових та прогресивних вебзастосунків (React, AngularJS, Vue.js).
3. Програмування на боці сервера (Node.js (Express.js)).
4. Стаціонарних застосунків (Electron , NW.js).
5. Мобільних застосунків (React Native, Cordova).
6. Сценаріїв в прикладних програмах (наприклад, в програмах зі складу Adobe Creative Suite чи Apache JMeter).

Сучасна веброзробка передбачає використання як стандартного JavaScript (Vanilla JavaScript), так і спеціалізованих бібліотек та фреймворків. Для невеликих вебпроектів і лендінгових сторінок можливостей стандартного JavaScript зазвичай достатньо. Водночас під час створення складних вебдодатків активно використовуються спеціалізовані інструменти, які спрощують розробку та підтримку програмного коду.

JavaScript-бібліотека – це колекція JS коду, який надає веброзробнику готові рішення для повсякденних задач. JavaScript-фреймворк також надає готові рішення, але також накладає певний стиль програмування [20]. Найпопулярнішими фреймворками та бібліотеками є: React, Angular, Vue.js, jQuery. Ступінь їхньої популярності є предметом постійної аналітики моніторингових груп. Згідно аналітичних звітів компанії W3Tech [19], 17,8% вебсайтів не використовують жодну з бібліотек JavaScript, які потрапили в зону опитування, jQuery використовується на 77,7 % усіх вебсайтів (верхня смуга), що складає 94,6 % частки ринку бібліотек JavaScript.

Фреймворки (табл. 1.5) являють собою чітку структуру додатку і реалізуються з використанням так званих патернів проектування.

Таблиця 1.5 – Порівняння популярних JavaScript-технологій

Технологія	Тип	Основні переваги	Сфера застосування
React	Бібліотека	Компонентний підхід, висока продуктивність, велика спільнота	SPA, вебдодатки
Angular	Фреймворк	Повний набір інструментів для розробки	Корпоративні системи
Vue.js	Фреймворк	Простота освоєння, висока швидкодія	Середні та великі вебпроекти
jQuery	Бібліотека	Простота роботи з DOM	Підтримка існуючих проєктів

Найбільш широко використовувані патерни це: MVC (Model-View-Controller), MVP (Model-View-Persistor), MVVW (Model-View-ViewModel). Додатків на фреймворках доволі багато, і цей сегмент набирає темп [21].

Окрім клієнтської частини вебресурсу, існує серверна частина (backend), яка відповідає за обробку даних, взаємодію з базами даних, автентифікацію

користувачів та реалізацію бізнес-логіки. Для створення серверної частини використовуються різні мови програмування та програмні платформи (табл. 1.6) [22].

Таблиця 1.6 – Порівняльний аналіз мов програмування для backend

Назва	Переваги	Недоліки
PHP	Потужність і гнучкість, простота, велике ком'юніті, багато готових рішень	Менша захищеність порівняно з іншими мовами, неможливість створення десктопного застосунку
Ruby	Можливість використання лише необхідних бібліотек, швидкість розробки, надійний захист даних	Проблеми при роботі на Windows, мала кількість вакансій, використання великої кількості пам'яті
Python	Дуже легкий у вивченні, універсальність, дуже популярний, великий вибір готових рішень.	Повільність виконання, динамічна типізація, не сама зручна мова для мобільної розробки, місцями незручний синтаксис
Node.js	Швидкість виконання, кросплатформність, велика кількість інструментів	Важко писати великі проєкти, залежність від бібліотек
Java	Незалежність від платформи, запуск в «пісочниці», автоматичне управління пам'яттю застосунка	Низька швидкість, важкість написання коду, плата за комерційне використання

Для лендінгових вебсайтів, до яких належить і вебсайт молодіжного кафе, потреба у складній серверній логіці зазвичай відсутня. Основний акцент робиться на якісній реалізації інтерфейсу користувача та адаптивності вебресурсу.

JavaScript можна використовувати, наприклад, для створення форми замовлення компонентів, перевірки достовірності даних, введених користувачем, та передачі даних на сервер у проєкті вебсайту зі збирання ПК. За допомогою заданих функцій JavaScript можна проводити обрахунки даних та виводити результати на екран, не звертаючись до сервера. Крім того, JavaScript дозволяє взаємодіяти з API, що дає змогу отримувати найсвіжішу інформацію про компоненти безпосередньо з вебсайтів виробників.

Ще однією важливою особливістю JavaScript є можливість додавання на сайт динамічних елементів, таких як випадаючі меню, вкладки, повзунки тощо.

Це підвищить естетичну привабливість сайту та покращить взаємодію з користувачами. Таким чином, основою сучасної веброзробки є комплексне використання HTML, CSS та JavaScript. HTML забезпечує структурування контенту, CSS відповідає за його візуальне оформлення та адаптивність, а JavaScript реалізує інтерактивну взаємодію користувача з вебресурсом. Використання сучасних засобів адаптивної верстки, бібліотек та фреймворків дозволяє створювати ефективні вебсайти, орієнтовані на користувачів різних типів пристроїв.

1.3 Характеристики дизайну вебсайтів у контексті сучасних трендів

Від того, наскільки грамотно буде оформлений вебсайт, буде залежати кількість користувачів і репутація організації. Важливо, щоб дизайн відображав напрямки діяльності компанії. Наприклад, якщо це сайт організації свят, то і їх домен повинен виглядати яскраво. А якщо це сайт банку, то використання яскравих кольорів і великих зображень буде недоречним.

Дизайн сучасних вебсайтів прийнято розглядати з позицій не лише естетичної привабливості, але й зручності використання, доступності та ефективності взаємодії користувача з інформаційним ресурсом. При проєктуванні вебінтерфейсів особлива увага приділяється концепціям UI (User Interface) та UX (User Experience), які є основою сучасного вебдизайну.

UI (User Interface) – це користувацький інтерфейс, який охоплює всі візуальні елементи взаємодії користувача із вебресурсом. До таких елементів належать кнопки, меню навігації, форми введення даних, іконки, зображення, шрифти, кольорові схеми та інші компоненти інтерфейсу. Основним завданням UI-дизайну є створення візуально привабливого, логічного та зрозумілого середовища для користувача.

UX (User Experience) характеризує досвід взаємодії користувача із вебсайтом та визначає рівень зручності використання ресурсу. UX-дизайн

охоплює структуру сторінок, логіку навігації, швидкість пошуку необхідної інформації, зрозумілість елементів керування та загальне сприйняття вебсайту користувачем. Якісний UX-дизайн дозволяє мінімізувати кількість дій, необхідних для досягнення цільової мети, та підвищує рівень задоволеності користувачів [23].

UI та UX є взаємопов'язаними складовими сучасного вебдизайну. Навіть візуально привабливий сайт може виявитися неефективним за наявності складної навігації або нелогічної структури. Водночас зручний, але застарілий інтерфейс може негативно впливати на сприйняття бренду. Сучасний підхід до проєктування вебресурсів передбачає комплексне врахування обох складових.

Дизайн вебсайту є важливим інструментом формування іміджу організації та залучення цільової аудиторії. Вдале дизайнерське рішення повинно відповідати специфіці діяльності компанії, її позиціонуванню на ринку та очікуванням потенційних користувачів. Саме тому дизайн банківських установ, освітніх платформ, розважальних сервісів або закладів громадського харчування суттєво відрізняється за кольоровими рішеннями, композицією та характером подання інформації.

Попри постійну зміну трендів, можна виділити кілька ключових тенденцій вебдизайну, які залишаються актуальними протягом останніх років:

- адаптивність та mobile-first проєктування [24];
- мінімалізм і функціональність інтерфейсу;
- використання великих якісних зображень;
- акцентна типографіка;
- мікроанімації та інтерактивні ефекти;
- використання контрастних кольорових акцентів;
- швидкість завантаження та оптимізація інтерфейсу;
- сторітелінг та емоційне представлення інформації [25];

Особливо важливим трендом є mobile-first підхід, який передбачає проєктування інтерфейсу насамперед для мобільних пристроїв із подальшим масштабуванням для більших екранів. Такий підхід пояснюється тим, що

значна частина користувачів отримує доступ до вебресурсів саме через смартфони. Для вебсайтів кафе та ресторанів цей показник є особливо високим, оскільки користувачі часто переглядають інформацію про заклад безпосередньо під час пошуку місця для відпочинку.

Одним із найпоширеніших напрямів сучасного вебдизайну залишається мінімалізм. Його основними принципами є простота композиції, відсутність надлишкових елементів, раціональне використання кольорів та акцентування уваги користувача на ключовому контенті. Мінімалістичний дизайн дозволяє покращити читабельність сторінок, пришвидшити завантаження вебресурсу та зробити навігацію більш зрозумілою для відвідувачів [26]. Кольори використовуються прості і в невеликій кількості. Мінімалізм чудово підходить для сайтів таких напрямків, як: IT, Travel, Fashion, портфоліо [27] (рис. 1.4). Цей стиль підходить і для завдання ресторанного бізнесу – дозволяє сконцентруватися на продукті. Крім того, він добре поєднується з концепцією лендінгових сторінок, де важливо забезпечити послідовне ознайомлення відвідувача з інформацією та спонукати до виконання цільової дії.

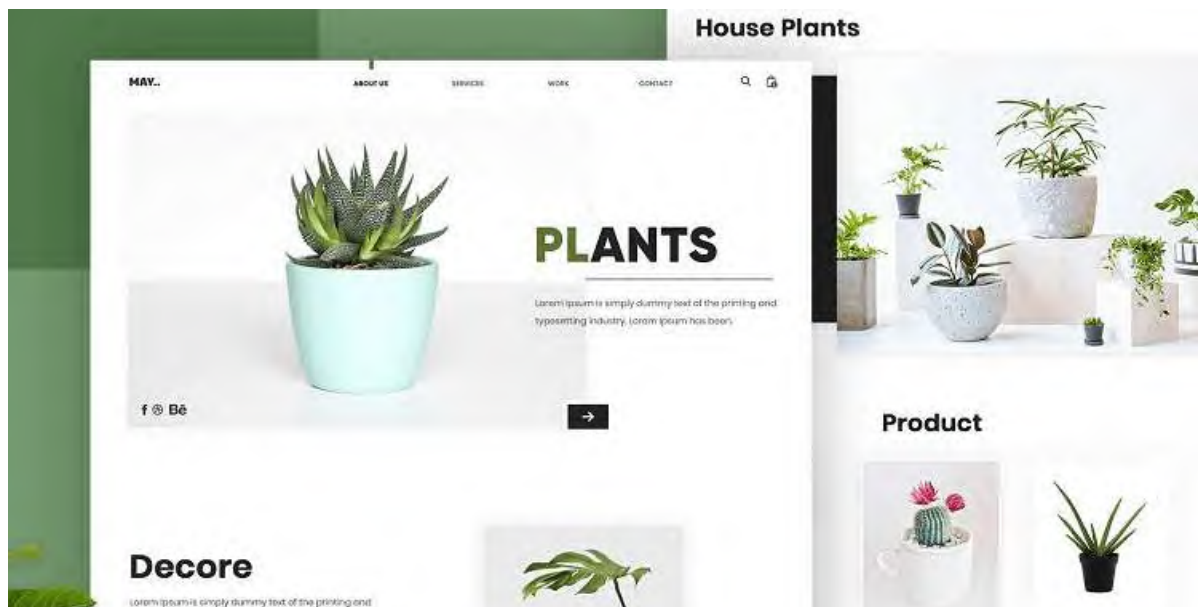


Рисунок 1.4 – Фрагмент вебсторінки з дизайном у мінімалістичному стилі

Для кращого розуміння особливостей сучасних стилів вебдизайну доцільно провести їх порівняння (табл. 1.7).

Таблиця 1.7 – Основні характеристики сучасних стилів вебдизайну

Стиль	Основні характеристики	Сфера застосування
Класичний	Стримані кольори, традиційна структура, акцент на інформацію	Корпоративні сайти, фінансові установи
Мінімалізм	Простота, багато вільного простору, акцент на контент	Лендінги, портфоліо, кафе та ресторани
Flat Design	Плоскі елементи, контрастні кольори, відсутність об'ємних ефектів	Вебсервіси, мобільні інтерфейси
Material Design	Картковий інтерфейс, плавні анімації, логічна ієрархія	Вебдодатки, інформаційні системи
Неомінімалізм	Поєднання мінімалізму з анімацією та сучасною типографікою	Сучасні брендові вебсайти

Серед перелічених стилів для вебресурсів сфери громадського харчування найбільш доцільним є використання мінімалістичного або неомінімалістичного підходу. Такі рішення дозволяють зосередити увагу користувача на основному продукті – страві, напоях, атмосфері закладу та спеціальних пропозиціях.

Для молодіжної аудиторії важливе значення також мають візуальна привабливість ресурсу, динамічність інтерфейсу та наявність якісного мультимедійного контенту. Саме тому сучасні сайти кафе часто використовують великоформатні фотографії продукції, плавну анімацію, інтерактивні елементи та контрастні акцентні кольори, які формують позитивне емоційне сприйняття бренду.

З огляду на проведений аналіз сучасних тенденцій вебдизайну, для розроблення шаблону адаптивного вебсайту молодіжного кафе доцільним є використання мінімалістичного стилю з елементами сучасного неомінімалізму. Такий підхід забезпечує поєднання функціональності, швидкодії, адаптивності та візуальної привабливості вебресурсу, що відповідає очікуванням сучасних користувачів та особливостям цільової аудиторії закладу.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ШАБЛОНУ АДАПТИВНОГО ВЕБСАЙТУ МОЛОДІЖНОГО КАФЕ

2.1 Побудова концепції розроблення шаблону сайту для втілення бізнес-плану молодіжного кафе

У сучасних умовах цифрової економіки наявність власного вебсайту є важливою складовою успішного функціонування бізнесу. Вебресурс виступає не лише інформаційним майданчиком, а й ефективним інструментом маркетингових комунікацій, формування бренду та залучення нових клієнтів. Особливо актуальним це є для закладів громадського харчування, діяльність яких значною мірою залежить від впізнаваності бренду, репутації та якості взаємодії з відвідувачами.

Молодіжні кафе належать до категорії закладів, цільова аудиторія яких активно використовує цифрові технології для пошуку місць відпочинку, перегляду меню, ознайомлення з відгуками та отримання актуальної інформації про акції й спеціальні пропозиції. У зв'язку з цим вебсайт стає одним із ключових каналів комунікації між закладом та потенційними клієнтами..

Розвиток онлайн-присутності закладів громадського харчування суттєво прискорився під час пандемії COVID-19, коли значна частина взаємодії зі споживачами була перенесена в цифровий простір. Після пандемії тенденція до використання вебсайтів та соціальних мереж для просування закладів збереглася і продовжує посилюватися. Загалом у світі налічується понад 500-600 тисяч діючих кав'ярень (включно з великими мережами, локальними брендами та дрібними незалежними закладами). Статистика базується на перехресних даних глобального ринку кав'ярень (World Coffee Portal, Eurostat), маркетингових дослідженнях та аналітиці інструментів розробки (наприклад, даних BuiltWith про інтеграцію онлайн-замовлень і систем e-commerce) [28].

Нижче наведено розподіл світового ринку кав'ярень за рівнем їхньої цифрової присутності, наявності або відсутності вебсайту (рис. 2.1).

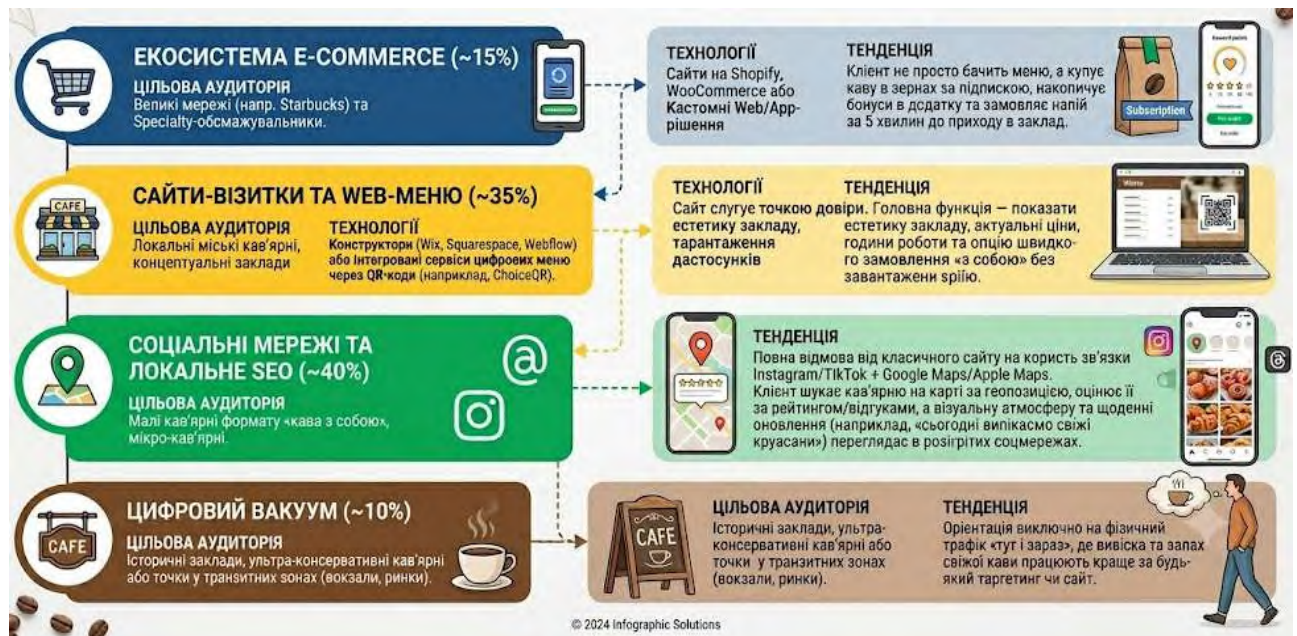


Рисунок. 2.1 – Сучасна екосистема інтернет-присутності кав'ярень у світі

Розподіл кав'ярень за типом вебсайту / цифрової присутності показує наступні тенденції (див. рис. 2.1).

1. Повноцінний e-commerce сайт (із доставкою/підпискою) - ~15%. Сюди входять великі мережі (Starbucks, Costa Coffee) та крафтові ростери (specialty coffee), які продають зерно онлайн, розвивають власні мобільні додатки та пропонують підписку на каву. Технології: сайти на Shopify, WooCommerce або кастомні Web/App-рішення.

2. Сайт-візитка з онлайн-меню або замовленням - ~35%. Локальні міські кав'ярні середнього рівня та локальні мережі. Сайт містить контакти, меню, карту проїзду та просту форму для замовлення «з собою». Технології: конструктори (Wix, Squarespace, Webflow) або інтегровані сервіси цифрових меню через QR-коди (наприклад, ChoiceQR) [29].

3. Соціальні мережі та Локальне SEO (без власного сайту) - ~40%. Малі кав'ярні формату «кава з собою», мікро-кав'ярні у спальних районах. Вони

повністю покладаються на Instagram, Facebook або локальні карти (Google Maps, Apple Maps).

4. Взагалі без цифрової присутності - ~10%. Традиційні старі європейські кав'ярні, кіоски у віддалених регіонах або заклади, орієнтовані суто на випадкових перехожих у транзитних зонах (вокзали, ринки). Орієнтація виключно на фізичний трафік «тут і зараз», де вивіска та запах свіжої кави працюють краще за будь-який таргетинг чи сайт.

При виконанні завдання кваліфікаційної роботи доцільно орієнтуватися на тенденції 1-2 пунктів, які забезпечені на 45% відповідними сайтами.

Метою даного проекту є розроблення універсального шаблону адаптивного вебсайту у формі лендінгу для молодіжного кафе. Шаблон повинен забезпечувати ефективне представлення бренду закладу, бути придатним для подальшої адаптації під різні концепції кафе та відповідати сучасним вимогам вебдизайну і користувацького досвіду.

Вибір саме шаблонного рішення обумовлений тим, що значна частина підприємств малого бізнесу не має можливості замовляти індивідуальну розробку вебресурсу. Універсальний шаблон дозволяє суттєво скоротити витрати на створення сайту та прискорити його впровадження, забезпечуючи при цьому достатній рівень функціональності та візуальної привабливості.

Під час проектування було обрано формат Landing Page (лендінг), який являє собою односторінковий вебресурс із послідовною подачею інформації. Такий формат є особливо ефективним для закладів громадського харчування, оскільки дозволяє швидко ознайомити користувача з основною інформацією про кафе, продемонструвати асортимент продукції, атмосферу закладу та мотивувати до здійснення цільової дії [30].

Мета полягає в тому, щоб зацікавити потенційного покупця протягом усього процесу продажу, включаючи зацікавлення, надання інформації, аргументацію та, у кінцевому підсумку, спонукати до покупки.

Основними перевагами використання лендінгу для молодіжного кафе є:

1. Концентрація всієї ключової інформації на одній сторінці.

2. Простота навігації та швидкий доступ до необхідних відомостей.
3. Високий рівень адаптивності для мобільних пристроїв.
4. Менша вартість розроблення та підтримки порівняно з багатосторінковими сайтами.
5. Високий потенціал конверсії завдяки послідовному ознайомленню користувача з контентом.
6. Можливість ефективного використання рекламних кампаній та просування у соціальних мережах.

Концептуальна структура односторінкового сайту з системою навігації за окремими розділами показана на рис. 2.2. Для кав'ярні це означає ефективну подачу бренду, швидкий доступ до основної інформації та зручність для користувачів, які найчастіше переглядають сайт з телефону.



Рисунок 2.2 – Концептуальна схема односторінкового сайту для кафе

З урахуванням особливостей цільової аудиторії було сформовано основні вимоги до майбутнього шаблону вебсайту:

- адаптивність для смартфонів, планшетів та персональних комп'ютерів;
- сучасний мінімалістичний дизайн із яскравими акцентами;
- використання великих якісних фотографій продукції та інтер'єру [31];
- зрозуміла навігація між розділами сторінки;

- швидке завантаження сторінок;
- використання анімацій та інтерактивних елементів для підвищення залученості користувачів [32];
- наявність контактної інформації та інтеграції із соціальними мережами;
- можливість подальшої адаптації шаблону для інших закладів громадського харчування.

З метою підвищення ефективності сприйняття інформації доцільно використовувати принципи сучасного мінімалістичного дизайну та неомінімалізму. Такий підхід передбачає використання достатньої кількості вільного простору, чіткої типографіки, контрастних акцентних кольорів та якісного візуального контенту. Особливу увагу планується приділити першому екрану вебсайту, який повинен формувати позитивне перше враження та стимулювати користувача до подальшого перегляду сторінки.

Таким чином, концепція розроблюваного шаблону передбачає створення сучасного адаптивного лендінгу для молодіжного кафе, який поєднує простоту використання, ефективну презентацію бренду та відповідність актуальним тенденціям вебдизайну.

2.2 Обґрунтування вибору методів та середовища створення сайту

Попередньо було з'ясовано (див. п.1.2), що на сучасному етапі існує значна кількість підходів до розроблення вебресурсів, серед яких використання конструкторів сайтів, готових шаблонів, систем керування контентом (CMS) та створення вебсайту шляхом самостійного написання програмного коду. Кожен із підходів має власні переваги та обмеження, тому для реалізації поставлених завдань було проведено їх порівняльний аналіз [33]. Для попереднього аналізу розглядалися популярні платформи-конструктори вебсайтів, зокрема Site123, Wix та інші аналогічні рішення, які надають готові шаблони для різних

напрямів бізнесу, включаючи заклади громадського харчування. Порівняння основних підходів до створення вебсайту наведено в табл. 2.2.

Таблиця 2.2 – Порівняння переваг основних існуючих підходів до розробки промосайту для бізнесу

Написання коду власноруч		Використання шаблону вебсайту	
Перевага	Зміст переваги	Перевага	Зміст переваги
Повна свобода дизайну	Ти сам вирішуєш, як саме має виглядати сайт – шрифти, анімації, колірна палітра (напр. «метро-естетика» - неон, плитка, індустріальні елементи).	Швидкість розгортання	Можна зробити сайт за день-два без глибоких знань коду.
Оптимізація продуктивності	Без зайвого коду, бібліотек або скриптів — сайт завантажуватиметься швидше, що важливо для SEO та мобільних користувачів	Менше технічної роботи	Не треба думати про адаптивність, кросбраузерність, SEO-шаблони зазвичай уже це мають.
Вдосконалення власної майстерності (навчання)	Якщо ти хочеш поглибити навички фронтенду чи UX/UI – самостійна розробка дає найбільше досвіду	Підтримка інтеграцій	Багато платформ підтримують додавання форм, аналітики, соцмереж прямо з коробки
Гнучкість у майбутньому	Легше змінювати логіку, підключати інші сервіси (наприклад, замовлення кави онлайн, бронювання місць тощо)	Візуальний редактор	Можна швидко експериментувати з виглядом без перезапуску коду
Індивідуальний характер сайту	Готові шаблони часто мають «типовий вигляд», а свій код – це унікальність, яка може краще передати дух кав'ярні.	Менша вартість	Менші витрати на отримання готового продукту через масовість, серійність

З огляду на це для реалізації проєкту було обрано підхід власного розроблення шаблону вебсайту. Такий підхід забезпечує повний контроль над структурою сторінок, дизайном інтерфейсу, адаптивністю та програмною реалізацією окремих функціональних елементів.

Враховуючи поставлену мету створення адаптивного лендінгу молодіжного кафе, як основні технології розроблення було обрано HTML5 [34], CSS3[35] та JavaScript. HTML5 використовується для формування логічної структури сторінок вебсайту та забезпечує коректне відображення

контенту в сучасних браузерях. Використання семантичних тегів HTML5 позитивно впливає на доступність ресурсу та його пошукову оптимізацію.

CSS3 застосовується для створення візуального оформлення вебсайту, реалізації адаптивної верстки та забезпечення єдиного стилю оформлення всіх елементів інтерфейсу. Використання сучасних можливостей CSS3, зокрема Flexbox та Grid Layout, дозволяє ефективно реалізувати адаптацію сторінок до різних розмірів екранів.

Для реалізації інтерактивних елементів обрано мову програмування JavaScript, який дозволяє створювати динамічні компоненти інтерфейсу, анімаційні ефекти, плавну навігацію між секціями лендінгу, а також забезпечує роботу без необхідності перезавантаження сторінки.

Оскільки розроблюваний вебсайт є відносно невеликим інформаційним ресурсом із обмеженим функціоналом, використання складних JavaScript-фреймворків, таких як React або Angular, було визнано недоцільним.

Для проєктування інтерфейсу та створення графічного прототипу вебсайту було використано програмне середовище Figma [36]. Дане рішення є одним із найбільш популярних інструментів UI/UX-дизайну та забезпечує можливість швидкого створення макетів, прототипування інтерфейсів і перевірки дизайнерських рішень до початку програмної реалізації.

Безпосереднє написання програмного коду виконувалося у середовищі Visual Studio Code [37]. Вибір цього редактора обумовлений його широкою підтримкою вебтехнологій, наявністю інструментів автоматичного форматування коду, інтеграцією із системами контролю версій та великою кількістю розширень для фронтенд-розробк

Visual Studio Code це сучасне інтегроване середовище розробки (IDE), створене компанією Microsoft для розробки GUI (графічного інтерфейсу користувача), консолі, вебпрограм, мобільноорієнтованих додатків, хмарних і вебсервісів тощо [37]. За допомогою цього IDE можна створювати керований код, а також рідний код. Він використовує різні платформи програмного забезпечення Microsoft для розробки програмного забезпечення (ПЗ). Це IDE

можна використовувати для написання коду на мовах C#, C++, VB (Visual Basic), Python, JavaScript і багато інших мов.

Основна відмінність між IDE та редакторами JavaScript полягає в тому, що останній не має багатьох функцій налагодження. Крім того, IDE підтримують різні системи ALM, такі як Git або GitHub. Однак, оскільки більшість редакторів коду підтримують ці системи, у цьому відношенні різниця поступово згладжується. У багатьох відношеннях IDE для розробки JavaScript є чудовими. Вони прискорюють процес розробки та роблять його ефективнішим. Крім того, їх здатність миттєво тестувати речі приносить численні переваги. Однак, незважаючи на це, редактори коду все ще популярні, оскільки багато з них ви можете отримати безкоштовно. Вони також легші, швидші та менш ресурсномісткі, тому їх можна використовувати на ноутбуках, смартфонах. Крім того, щоб зробити речі ще більш заплутаними, частина функціональних можливостей IDE вводиться в редактори коду. Це перетворює їх із розкішних блокнотів (вони були 10 років тому) на IDE-подібні платформи. Один із рейтингів IDE [38] для веброзробки наведено на рис. 2.3.

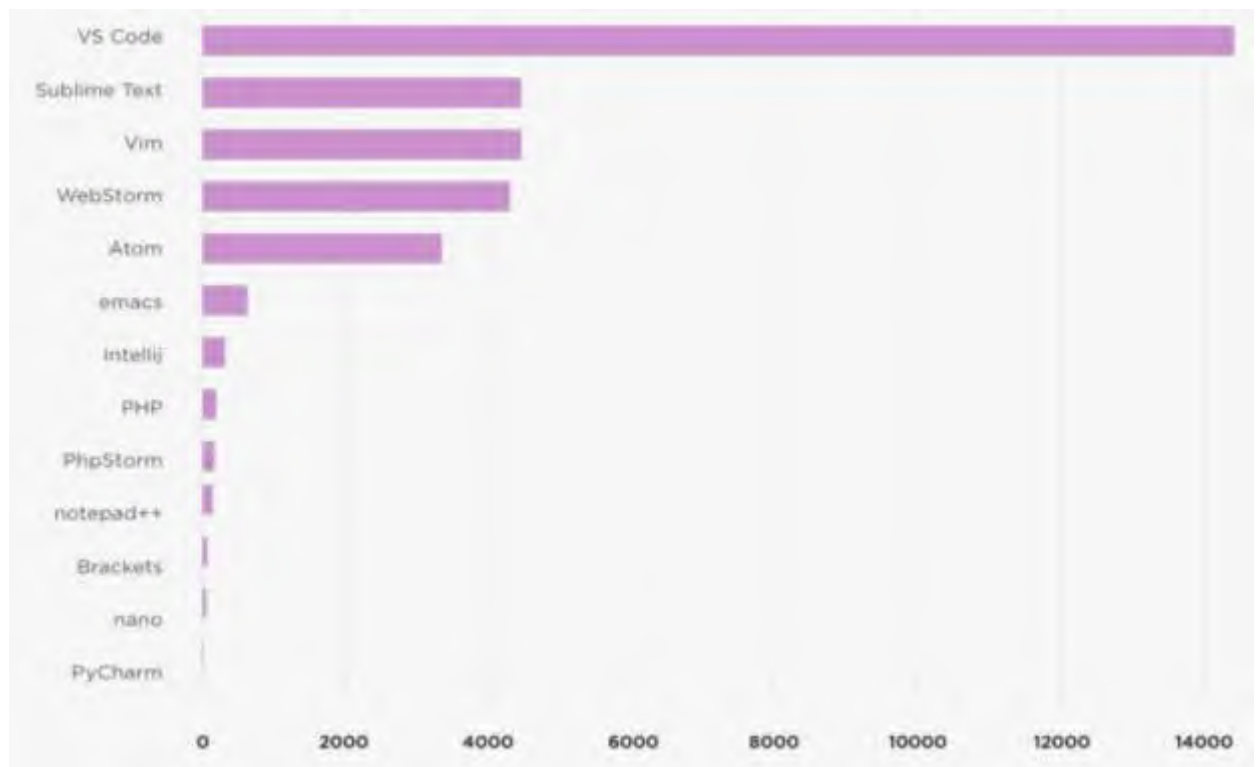


Рисунок 2.3 – Діаграма популярності редакторів коду та IDE для JS [38]

За даними моніторингів (див. рис. 2.3) першість серед IDE для JS тримає середовище програмування і розробки Visual Studio Code. За ним йдуть SublimeText, Vim, WebStorm і Atom. Visual Studio Code (VSC) має понад 14 тис. користувачів із 20 тис. учасників опитування [38]. Окрім JS, він підтримує 30 мов програмування, таких як Ruby, Python, C# тощо.

Підсвічування синтаксису, автозавершення, налагодження коду, взаємодія з системами контролю версій та багато інших важливих функцій доступні у Visual Studio Code. Ці функції допомагають прискорити розробку та підвищити якість коду.

Visual Studio Code – це практичний та ефективний інструмент для розробки вебзастосунків. Завдяки широким можливостям, функціональності та зручному інтерфейсу процес розробки є більш ефективним та результативним. Крім того, VSC має відкритий вихідний код і є безкоштовним, що робить його доступним для всіх користувачів. До VSC також включено потужні можливості налагодження коду, що дозволяє швидко знаходити та усувати помилки. Завдяки вбудованому відладчику ви можете розміщувати точки зупинки у вашому коді та досліджувати кожну стадію виконання програми.

Крім того, Visual Studio Code пропонує потужну екосистему розширень, які можна використовувати для персоналізації та розширення можливостей редактора. Існують розширення для роботи з різними мовами програмування, інтеграції з системами контролю версій, інструменти для підвищення продуктивності та багато іншого.

Використання Visual Studio Code для розробки вебзастосунків має кілька переваг: містить багато можливостей для налаштування та розширення функцій, є безкоштовним і з відкритим вихідним кодом, а також дозволяє легко створювати та редагувати код. Тому використання Visual Studio Code у нашому проєкті виявилось корисним і успішним. Це дозволило створити високоякісний код і успішно завершити проєкт.

Для створення інтерактивності на вебсторінці заплановано використовувати JavaScript, який є безумовним лідером серед мов

програмування для веброзробки [39]. Про популярність JS свідчить рейтинг мов програмування (рис.2.4).

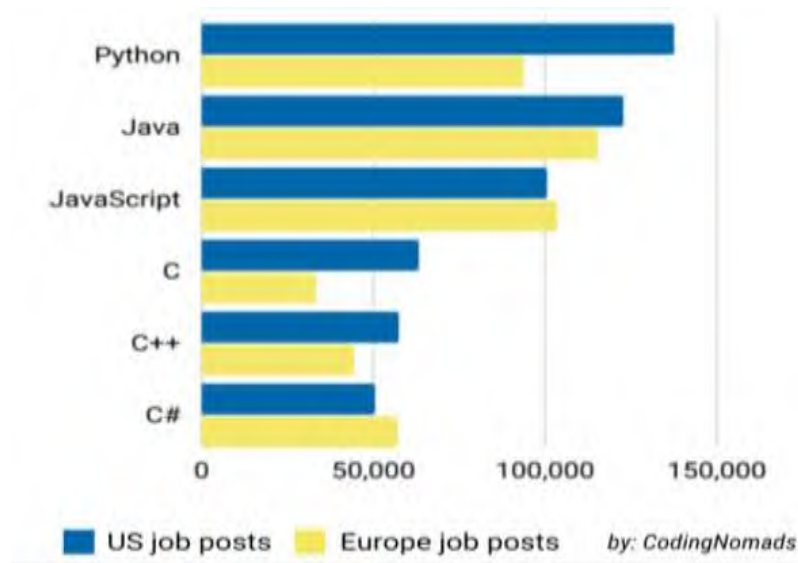


Рисунок 2.4 – Затребуваність мов програмування у 2025 р. (за даними [39])

До мінусів JS можна віднести те, що він не завжди коректно працює з різними браузерами, а також, оскільки JS виконується на стороні клієнта, то існує більша ймовірність злому клієнтських даних.

Отже, обраний технологічний стек HTML5, CSS3, JavaScript, Figma та Visual Studio Code повністю відповідає вимогам проєкту та забезпечує можливість створення сучасного адаптивного шаблону вебсайту молодіжного кафе у формі лендінгу.

2.3 Проєктування макету шаблону сайту з використанням Figma

Після визначення концепції майбутнього вебресурсу та обґрунтування вибору технологій розроблення наступним етапом стало проєктування інтерфейсу та створення макету шаблону вебсайту молодіжного кафе. Для виконання даного завдання було використано середовище Figma – сучасний хмарний інструмент для створення користувацьких інтерфейсів, прототипів та

дизайнерських макетів вебзастосунків [36]. Використання Figma дозволяє виконати попередню візуалізацію майбутнього вебсайту ще до початку програмної реалізації, оцінити зручність структури, логіку розташування елементів та відповідність дизайну поставленим бізнес-завданням. Крім того, застосування графічного прототипування дозволяє зменшити кількість помилок на етапі верстки та прискорити процес розроблення. До переваг Figma можна віднести наступні: вся робота ведеться в хмарі, зручний режим для розробників, безкшотовність. Крім того, вона дозволяє працювати в команді в режимі реального часу, що полегшує співпрацю дизайнерів і розробників, а також дозволяє клієнтам вносити зміни і коментарі безпосередньо в макеті. Середовище Figma пропонує різноманітні інструменти для створення фігур, ліній і форм, а також для накладання тексту, вибору з декількох кольорових палітр, додавання піктограм і багато іншого.

На початковому етапі проєктування було сформовано загальну концепцію вебсайту молодіжного кафе. Розроблення макету здійснювалося відповідно до структури лендінгу, визначеної у попередньому підрозділі. При цьому було використано блочний принцип побудови сторінки, за якого кожен тематичний розділ є окремим інформаційним блоком із власним змістовим наповненням. Робота з головним рисунком для верхньої частини вебсайту показана на рис. 2.5.

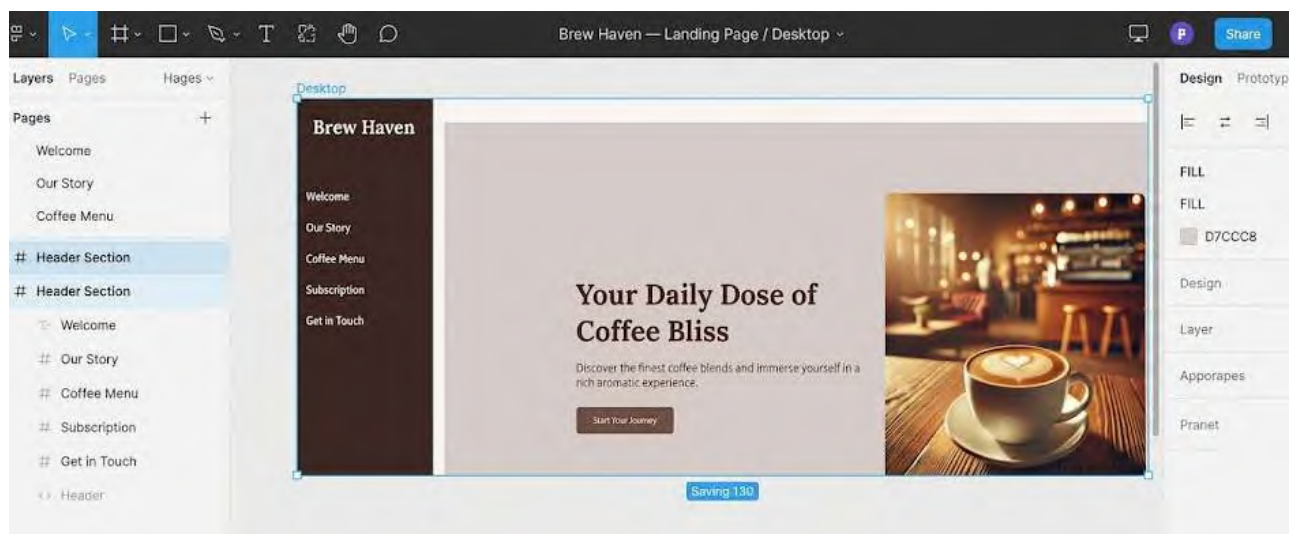


Рисунок 2.5 – Фрагмент створення макету в Figma

Результат роботи над верхньою частиною головної сторінки вебсайту після макетування у Figma показано на рис. 2.6.



Рисунок 2.6 – Візуальне зображення дизайну макету сайту (верхньої частини)

Перший екран виконує функцію привернення уваги відвідувача та містить ключову інформацію про заклад, слоган, головне зображення і кнопку швидкого переходу до меню або форми замовлення. Особлива увага була приділена розробленню центральної частини сторінки, у якій розміщується меню кафе та візуальна презентація основних позицій асортименту. Використання великих якісних фотографій дозволяє підвищити привабливість пропозиції та стимулювати зацікавленість потенційних клієнтів. Структурно макет вебсайту складається з таких основних розділів:

- головний екран із логотипом, навігаційним меню, слоганом та закликом до дії;
- розділ «Про кафе», який знайомить відвідувачів з концепцією закладу та його атмосферою;
- блок «Наша історія», де висвітлюються особливості створення бренду та його цінності;
- розділ меню із фотографіями та коротким описом напоїв і десертів;
- галерея інтер'єру та продукції закладу;
- блок відгуків клієнтів;

- форма замовлення або бронювання;
- інформаційний блок із контактними даними та посиланнями на соціальні мережі.

Оскільки обрано модель лендінгу, то заплановано кілька груп горизонтальними блоками і бокове меню. Основний зміст – меню кав'ярні з відповідними фото заплановано в середині сайту (рис. 2.7).

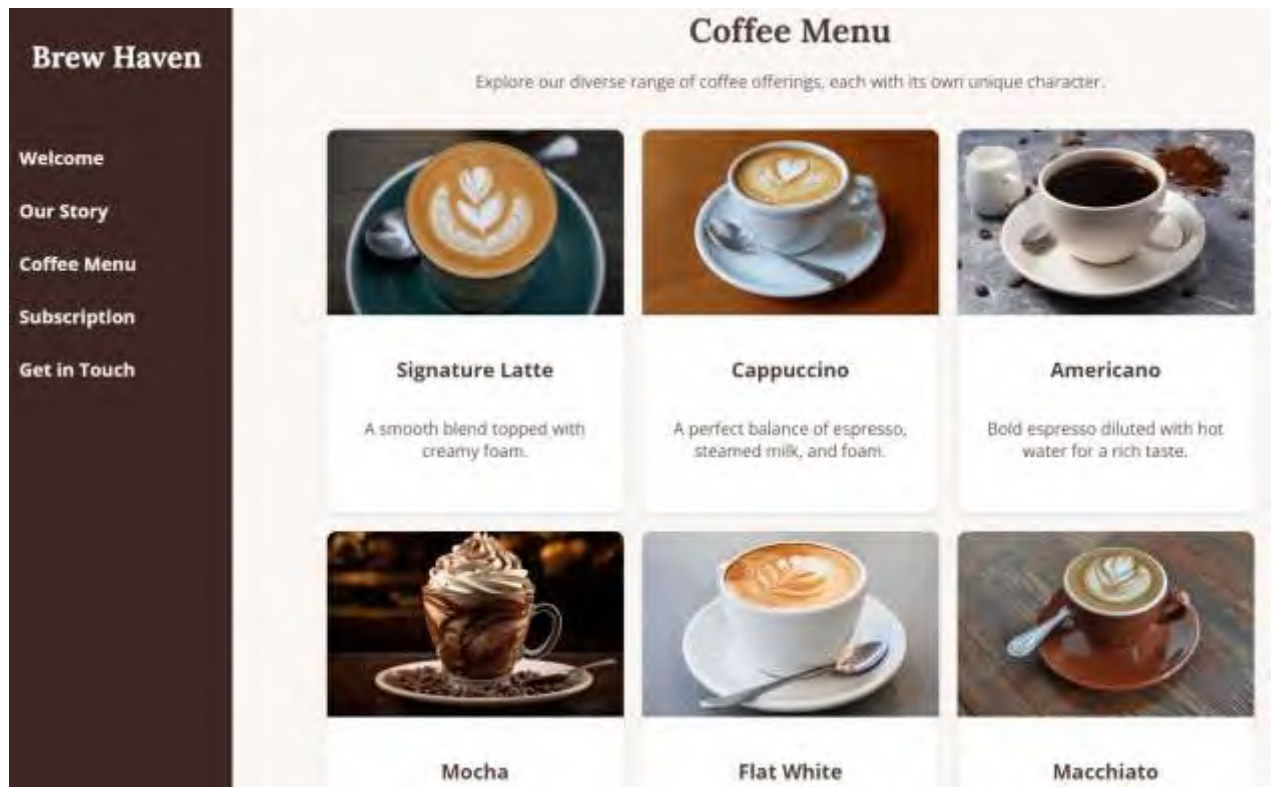


Рисунок 2.7 – Дизайн середньої змістової частини сайту молодіжного кафе

Під час проєктування особлива увага приділялася адаптивності майбутнього інтерфейсу. Макет було спроектовано таким чином, щоб усі елементи коректно відображалися як на персональних комп'ютерах, так і на планшетах та смартфонах. Це відповідає сучасним тенденціям веброзробки та особливостям поведінки цільової аудиторії молодіжних закладів харчування.

Процес дизайну сайту в Figma складається з наступних етапів:

1. Створення рамки проєкту – визначення обсягу робіт, вибір кольорової гами, шрифтів та інших елементів. Найкращим колористичним рішенням є

кольори натуральної кави з молоком. Гармонійність такого рішення видно на рис. 2.7-2.8, де представлена верхня і середня частини сайту.

2. Розмітка – створення загальної структури сторінки, блоків.
3. Додавання елементів дизайну – тексту, зображень, кнопок та інших.
4. Взаємодія з елементами – додавання анімації та інтерактивності, щоб зробити дизайн більш динамічним та привабливим для користувачів.
5. Прототипування – створення прототипу, щоб показати, як буде виглядати та працювати сайт на різних екранах та пристроях.
6. Тестування – перевірка функціональності та коректності відображення дизайну на різних пристроях та екранах.

Повний дизайн розробленого макету в графічному середовищі представлено в додатку А.

Важливою особливістю розробленого шаблону є використання англomовного інтерфейсу користувача. Таке рішення було прийнято з огляду на універсальність майбутнього вебресурсу та можливість його використання не лише в межах локального ринку, але й для міжнародних проєктів ресторанного бізнесу. Англійська мова є загальноприйнятим стандартом у сфері веброзробки та цифрового маркетингу, що дозволяє адаптувати створений шаблон для закладів харчування в різних країнах без суттєвої зміни структури інтерфейсу. Крім того, використання англomовних елементів спрощує подальшу локалізацію сайту іншими мовами та розширює потенційну аудиторію користувачів.

Отже, результатом виконаного етапу стало створення повноцінного графічного прототипу адаптивного вебсайту молодіжного кафе, який відображає структуру майбутнього ресурсу, його стильове оформлення та особливості взаємодії користувача з інтерфейсом. Створений макет став основою для подальшої програмної реалізації вебсайту засобами HTML5, CSS3 та JavaScript.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЦЕСУ РОЗРОБЛЕННЯ ПРОМО-САЙТУ ДЛЯ СПЕЦІАЛІЗОВАНОЇ КАВ'ЯРНІ

3.1 Розроблення основного коду промо-сайту і забезпечення функцій

Як було обґрунтовано в попередніх розділах, в якості технологій для написання коду та верстки були використані мови HTML, CSS та JS разом із середовищем розробки Visual Studio Code, де і починається робота над програмною реалізацією сайту (рис. 3.1). На початку потрібно створити папку проєкту, визначитись і створити структуру файлів із заданими типами. За потреби необхідно підключити плагіни, такі як: автоформатування коду, Emmet – для легшого написання HTML та CSS, Live Server для роботи с JavaScript. Фрагмент робочої області на початку роботи наведено на рис. 3.1.

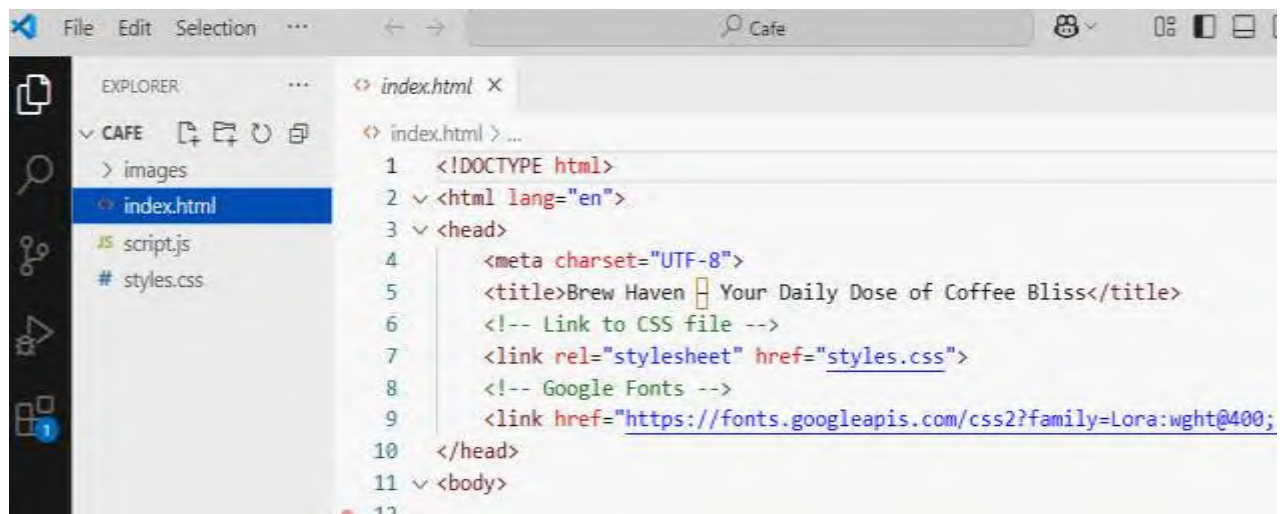


Рисунок 3.1 – Зміст HTML-файлу в середовищі VS Code і структура проєкту

Довжина коду HTML сторінки складає 160 рядків з тегами (додаток Б). Для розмітки структури використані як універсальні блокові елементи <div>, наприклад <div class="content">, так і семантичні теги <section>, <footer>, <nav>. Серед оригінальних рішень в коді є блок бокового меню, яке залишається на сторінці постійно. Код розмітки меню показане на рис. 3.2.

```

14 <nav id="side-nav">
15   <div class="logo">
16     <h1>Brew Haven</h1>
17   </div>
18   <ul>
19     <li><a href="#home">Welcome</a></li>
20     <li><a href="#story">Our Story</a></li>
21     <li><a href="#menu">Coffee Menu</a></li>
22     <li><a href="#subscription">Subscription</a></li>
23     <li><a href="#contact">Get in Touch</a></li>
24   </ul>
25   <div class="social-media">
26     <a href="#"></a>
27     <a href="#"></a>
28     <a href="#"></a>
29   </div>
30 </nav>

```

Рисунок 3.2 – Розмітка блоку меню з назвою та посиланнями на соцмережі

Після розмітки структура всього HTML-документу необхідно додавати стилі CSS, щоб зробити сайт більш технологічним, функціональним і відповідним задуманому дизайну. На цьому кроці додаються стилі для різних елементів на сторінці, таких як шрифти, кольори, фони, рамки, трансформації тощо. Повний зміст файлу CSS наведено в додатку В. Опис властивостей меню (див. рис. 3.2) розміщено у відповідному ідентифікаторі #side-nav, який зв'язаний із тегом nav (рядок 14: <nav id="side-nav">) і представлено на рис. 3.3.

```

15 #side-nav {
16   position: fixed;
17   top: 0;
18   left: 0;
19   width: 250px;
20   height: 100%;
21   background-color: #3e2723;
22   display: flex;
23   flex-direction: column;
24   align-items: center;
25   padding-top: 20px;
26 }

```

Рисунок 3.3 – Перелік властивостей стилю блоку навігації в ідентифікаторі

Написання кодів HTML і CSS – це творчий процес, який вимагає розуміння того, які елементи потрібні на сторінці, і вміння працювати з

різними властивостями CSS. Наприклад, спеціальні інтерактивні властивості при активації одного з пунктів меню користувачем сайту шляхом наведення мишки на пункт меню (подія `a:hover`) і зміна кольору при такій події показані у фрагменті коду на рис. 3.4.

```

45 #side-nav ul li a {
46     display: block;
47     color: #faf7f5;
48     padding: 15px 20px;
49     text-decoration: none;
50     font-weight: bold;
51     transition: background-color 0.3s;
52 }
53
54 #side-nav ul li a:hover {
55     background-color: #5d4037;
56 }

```

Рисунок 3.4 – Фрагмент опису CSS-властивостей для пунктів меню при наведенні курсором та активації користувачем

Для цього ж ідентифікатора далі додані інші властивості, які визначають адаптивність та налаштування перегляду на малих екранах (розглянуто далі).

Наступний змістовний блок організований як три окремих секції з однаковим оформленням (рис. 3.5).

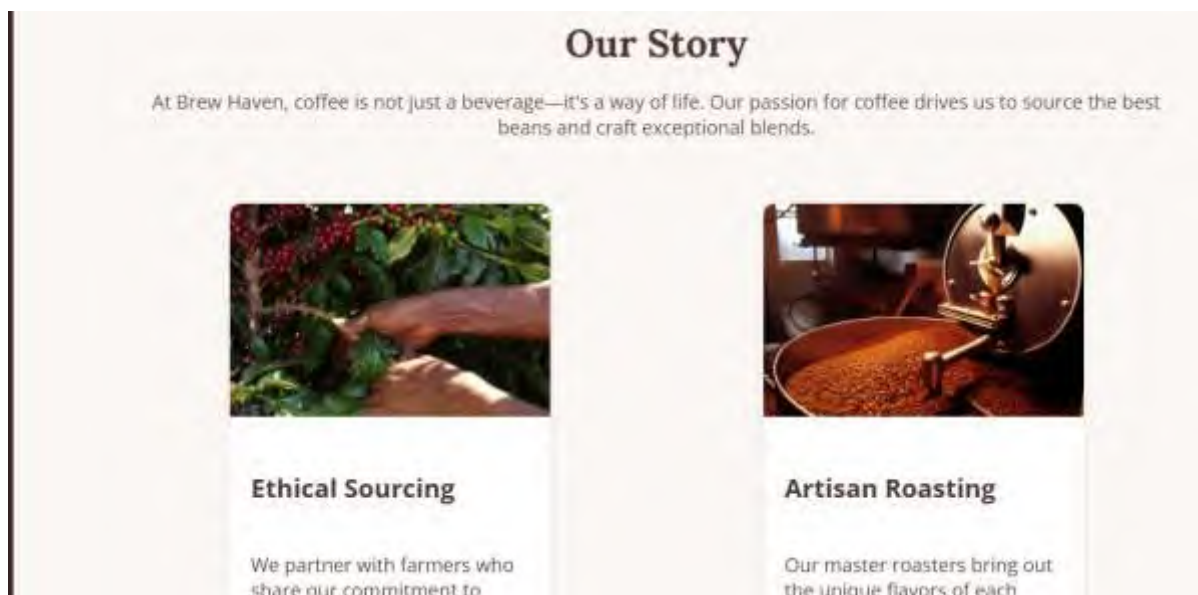


Рисунок 3.5 – Зовнішній вигляд частини шаблону про збір і купаж кави

Основні елементи секції задані блоками, які об'єднані в спільному блоці (рис. 3.6) з використанням однакових тегів `<div>`, і мають однакове оформлення із посиланням на той самий клас `“card”`(рис. 3.7).

```

47     <section id="story">
48         <div class="content">
49             <h2>Our Story</h2>
50             <p>At Brew Haven, coffee is not just a beverage—it's a way of life.
51                 Our passion for coffee drives us to source the best beans and craft exceptional blends.</p>
52             <div class="story-cards">
53                 <div class="card">
54                     
55                     <h3>Ethical Sourcing</h3>
56                     <p>We partner with farmers who share our commitment to quality and sustainability.</p>
57                 </div>
58                 <div class="card">
59                     
60                     <h3>Artisan Roasting</h3>
61                     <p>Our master roasters bring out the unique flavors of each bean.</p>
62                 </div>
63                 <div class="card">
64                     
65                     <h3>Coffee Community</h3>
66                     <p>Join a community of coffee lovers and share your experiences.</p>
67                 </div>
68             </div>
69         </div>
70     </section>

```

Рисунок 3.6 – Структура HTML коду для змістової частини сайту під заголовком «Our Story»

```

153     .card {
154         background-color: #fff;
155         width: 300px;
156         margin: 20px;
157         border-radius: 10px;
158         overflow: hidden;
159         text-align: left;
160         box-shadow: 0 2px 6px rgba(0,0,0,0.1);
161     }
162
163     .card img {
164         width: 100%;
165         height: 200px;
166         object-fit: cover;
167     }
168
169     .card h3 {
170         padding: 20px;
171         font-size: 24px;
172         color: #4e342e;
173     }
174
175     .card p {
176         padding: 0 20px 20px;
177         color: #5d4037;

```

Рисунок 3.7 – Зміст каскадних таблиць стилів у файлі `styles.css` для опису типових елементів, що належать класу `.card`

Завдяки каскадності стилів, код з описом CSS вийшов компактним і керованим. В основному класі `.card` для кожного з трьох блоків секції визначено форму, розмір, фон `background-color: #fff;`, відступи, заокруглену форму (`border-radius: 10px;`), а також тінь (`box-shadow: 0 2px 6px rgba(0,0,0,0.1);`) та деякі інші елементи.

Наступна секція побудована за таким же принципом: блок має заголовок і короткий текст, після чого слідує набір із 6 блоків з прикладами подачі кави за різною (в чомусь унікальною) рецептурою. Структура головного блоку показана на рис. 3.8, розмітка одного з блоків реклами показана на рис. 3.9.

```

72 <section id="menu">
73 <div class="content">
74 <h2>Coffee Menu</h2>
75 <p>Explore our diverse range of coffee offerings, each with its own unique character.</p>
76 <div class="menu-grid">

```

Рисунок 3.8 – Ієрархічна структура трьох трьох рівнів вкладених блоків

```

77 <div class="menu-item">
78 
79 <h3>Signature Latte</h3>
80 <p>A smooth blend topped with creamy foam.</p>
81 </div>

```

Рисунок 3.9 – Набір елементів одного з блоків рекламного типу в секції «Coffee Menu»

Всі блоки сайту оформлені за однаковим принципом, але кожен з блоків має оригінальне фото (рис. 3.10). Це дозволяє за потреби масштабувати розділ і збільшити кількість описових частин. Згідно розмітки (див. рис. 3.9) блок містить фото, тематичний заголовок (рівень `h3`), короткий описовий текст. Рисунок перекриває частину блоку. Всі зображення однакового розміру, відмінної якості. Заголовки мають однаковий стиль: колір і розмір шрифту, колір фону блоку, додаткові властивості – закруглення, тіні тощо. Код властивостей елементів, які належать класу `.menu-item`, показано на рис. 3.10.

```

187 .menu-item {
188     background-color: #fff;
189     border-radius: 10px;
190     overflow: hidden;
191     text-align: center;
192     box-shadow: 0 2px 6px rgba(0,0,0,0.1);
193 }
194
195 .menu-item img {
196     width: 100%;
197     height: 200px;
198     object-fit: cover;
199 }
200
201 .menu-item h3 {
202     padding: 15px;
203     font-size: 22px;
204     color: #4e342e;
205 }
206
207 .menu-item p {
208     padding: 0 15px 15px;
209     color: #5d4037;
210 }

```

Рисунок 3.10 – Властивості елементів, які належать класу .menu-item

Завдяки каскадності та оптимальному використанню CSS, досягнуто раціональний запис стилів без повторення та надмірних значень параметрів.

Для створення інтерактивних елементів вебсторінок використовується JavaScript, який є важливим компонентом веброзробки. Використовуючи цю мову програмування, можна додавати динамічні компоненти на вебсайти, щоб вони реагували на введення користувача без необхідності оновлення сторінки.

Це особливо корисно для нашого завдання, оскільки сайт повинен мати інтерактивні компоненти для динамічних операцій, наприклад, розгортання даних та анімація. Кнопки, меню, форми, таблиці, графіки, карти та інші інтерактивні компоненти можуть бути створені на сторінках також за допомогою JavaScript. В даній роботі створено окремий файл зі скриптами та посилання на нього в кінці розмітки HTML, за допомогою яких організовано скролінг та окремі динамічні ефекти.

Скролінг для лендінгу завжди актуальний і є основним показником зручності UX (взаємодії користувача). Необхідно було технічно добитися плавного прокручування сторінки. Відповідний код JS показано на рис. 3.11.

```

JS script.js > ...
1  /* Smooth scrolling adjusted for fixed navigation */
2  document.querySelectorAll('#side-nav ul li a').forEach(anchor => {
3      anchor.addEventListener('click', function(e) {
4          e.preventDefault();
5          const targetId = this.getAttribute('href').substring(1);
6          const targetSection = document.getElementById(targetId);
7
8          // Можна відрегулювати відступ за потреби
9          window.scrollTo({
10             top: targetSection.offsetTop - 20,
11             behavior: 'smooth'
12         });
13     });
14 });

```

Рисунок 3.11 – Зображення фрагменту скрипта JS для скролінгу при написанні в VS Code

Серед вибраних функцій плавність досягається за допомогою функції `behavior: 'smooth'`. При натисканні кнопки вниз блоку меню на сайті впливає повідомлення про можливість отримання всього меню через пошту. Технічно це реалізовано в сукупності трьох складових:

1. Розміщення і опис кнопки в HTML: `<button class="view-full-menu-button" onclick="showMessage('Our full menu has been sent to your email!')">View Full Menu</button>`.

2. Опис властивостей CSS кнопки показано на рис. 3.12.

```

212  /* View Full Menu Button */
213  .view-full-menu-button {
214      background-color: #8d6e63;
215      color: #faf7f5;
216      padding: 15px 30px;
217      border: none;
218      border-radius: 5px;
219      cursor: pointer;
220      transition: background-color 0.3s;
221      font-size: 18px;
222      font-weight: bold;
223      margin-top: 20px;
224      text-transform: uppercase;
225  }
226
227  .view-full-menu-button:hover {
228      background-color: #7b5e57;
229  }

```

Рисунок 3.12 – Опис властивостей кнопки для перегляду меню на сайті

3. Скрипти для динамічної поведінки повідомленням при обробці події кнопки представлені на рис. 3.13, візуальне зображення появи повідомлення з текстом – на рис. 3.14.

```

16  /* Function to display notification messages */
17  function showMessage(message) {
18      const container = document.getElementById('notification-container');
19
20      // Create a new notification element
21      const notification = document.createElement('div');
22      notification.classList.add('notification');
23      notification.textContent = message;
24
25      // Add notification to the container
26      container.appendChild(notification);
27
28      // Remove the notification after animation ends
29      setTimeout(() => {
30          notification.remove();
31      }, 5000); // 5 seconds
32  }

```

Рисунок 3.13 – Опис поведінки повідомлення на JavaScript

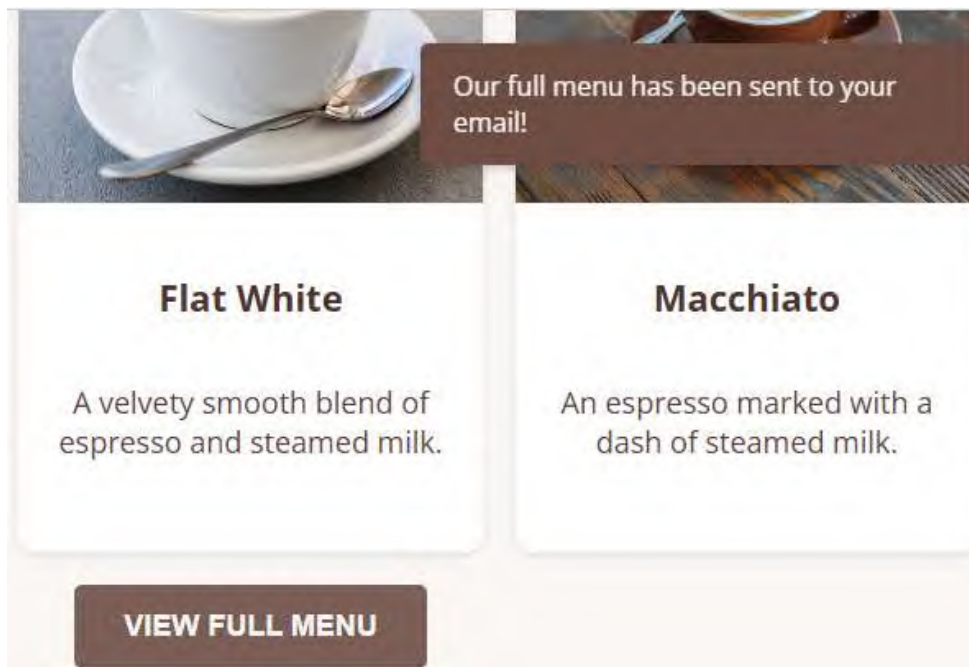


Рисунок 3.14 – Момент «впливання» повідомлення після натискання кнопки

Повідомлення показується на фоні блоків кавового меню і тримається встановлений час – 5 с (див. рис. 3.13), а потім плавно зникає.

3.2 Удосконалення результату розробки шаблону сайту для адаптивності на екранах смартфонів та елементи SEO-оптимізації

Більшість цільової аудиторії такого застосунку, як промо-сайт спеціалізованої кав'ярні, буде переглядати інформацію, робити замовлення або ж брати участь в акціях через смартфон. Тому неодмінним атрибутом сучасного вебсайту є його адаптивність. Для забезпечення гнучкості використання в таблицях стилів було додано відповідні властивості: для адаптивності меню `display: flex;` і `flex-direction: column;` (рис. 3.15). Перша відповідає за гнучкість, друга розміщує пункти вертикально, в колонку.

```

15  #side-nav {
16      position: fixed;
17      top: 0;
18      left: 0;
19      width: 250px;
20      height: 100%;
21      background-color: #3e2723;
22      display: flex;
23      flex-direction: column;
24      align-items: center;
25      padding-top: 20px;
26  }

```

Рисунок 3.15 – включення властивостей адаптивності в коді бокового меню

Так само, властивість `display: flex;` додана в клас `.hero`, який описує верхній блок сайту (рис. 3.16).

```

77  /* Hero Section */
78  .hero {
79      display: flex;
80      align-items: center;
81      justify-content: center;
82      background-color: #d7ccc8;
83      padding: 100px 20px;
84  }

```

Рисунок 3.16 – Фрагмент властивостей головного меню з адаптивним елементом

Для блоків другої секції так само задано `display: flex;`, а також функцію стискання внутрішніх блоків (рис. 3.17).

```

146  /* Story Cards */
147  .story-cards {
148      display: flex;
149      justify-content: space-around;
150      flex-wrap: wrap;
151  }

```

Рисунок 3.17 – Фрагмент опису властивостей адаптивного блоку

В нижній частині промо-сайту розміщено інформацію з контактами на фоні фото з краєвидом вулиці, де може бути локація кафе. Тут також встановлені необхідні властивості адаптивності та можливості зменшення розміру `flex-wrap: wrap;` (рис. 3.17).

```

282  /* Contact Info */
283  .contact-info {
284      display: flex;
285      justify-content: space-around;
286      flex-wrap: wrap;
287      margin-bottom: 40px;
288  }
289
290  .info-item {
291      display: flex;
292      align-items: center;
293      margin: 20px;
294  }

```

Рисунок 3.17 – Опис адаптивних властивостей в нижній секції промо-сайту

Отже, для розміщення елементів на сторінці використовується CSS flexbox. Це технологія для створення гнучких макетів за рахунок правильного розміщення елементів на сторінці. За допомогою неї можна легко керувати дочірніми елементами, міняти їх напрям, відстані між ними та багато іншого.

Для спрощення створення адаптиву, а також для того щоб весь контент сайту не розпливався по краям, створюється CSS-клас `.section`, в якому задається максимальна ширина, зокрема, в нашому випадку 100 % або 1000 px.

При роботі над проектом це значення можна буде змінювати в залежності від пристрою. До властивостей класу додається `flex`.

Окремо було доопрацьована адаптивність меню та перетворення в структуру гамбургер-меню для малих екранів. Відповідні елементи були додані в HTML-код (рис. 3.18), властивості CSS (рис. 3.19), а також в програмний код JavaScript (рис. 3.20).

```

16      <!-- Hamburger Button -->
17      <button id="hamburger" aria-label="Toggle Navigation">
18      ...<span class="bar"></span>
19      ...<span class="bar"></span>
20      ...<span class="bar"></span>
21      </button>
22      <h1>Brew Haven</h1>

```

Рисунок 3.18 – Вставка додаткових елементів гамбургер-меню в HTML

```

460  /* Hamburger Styles */
461  #hamburger {
462      display: none;
463      flex-direction: column;
464      justify-content: center;
465      gap: 5px;
466      background: none;
467      border: none;
468      cursor: pointer;
469      margin-bottom: 20px;
470  }
471
472  #hamburger .bar {
473      width: 25px;
474      height: 3px;
475      background-color: #faf7f5;
476      transition: all 0.3s;
477  }

```

Рисунок 3.19 – Додавання властивостей гамбургер-меню в ідентифікатор `#hamburger`

Зрештою, для забезпечення функціональності гамбургер-меню додається фрагмент програмного коду в кінець файлу `script.js` (рис. 3.20).

```
41  /* Hamburger menu toggle */  
42  const hamburger = document.getElementById('hamburger');  
43  const navLinks = document.querySelector('#side-nav ul');  
44  
45  hamburger.addEventListener('click', () => {  
46    navLinks.classList.toggle('active');  
47  });
```

Рисунок 3.20 – Зміст скрипта для згортання меню в «гамбургер»

При проведенні попереднього тестування розробленого шаблону отримали вигляд згорнутого меню, як показано на рис. 3.21.

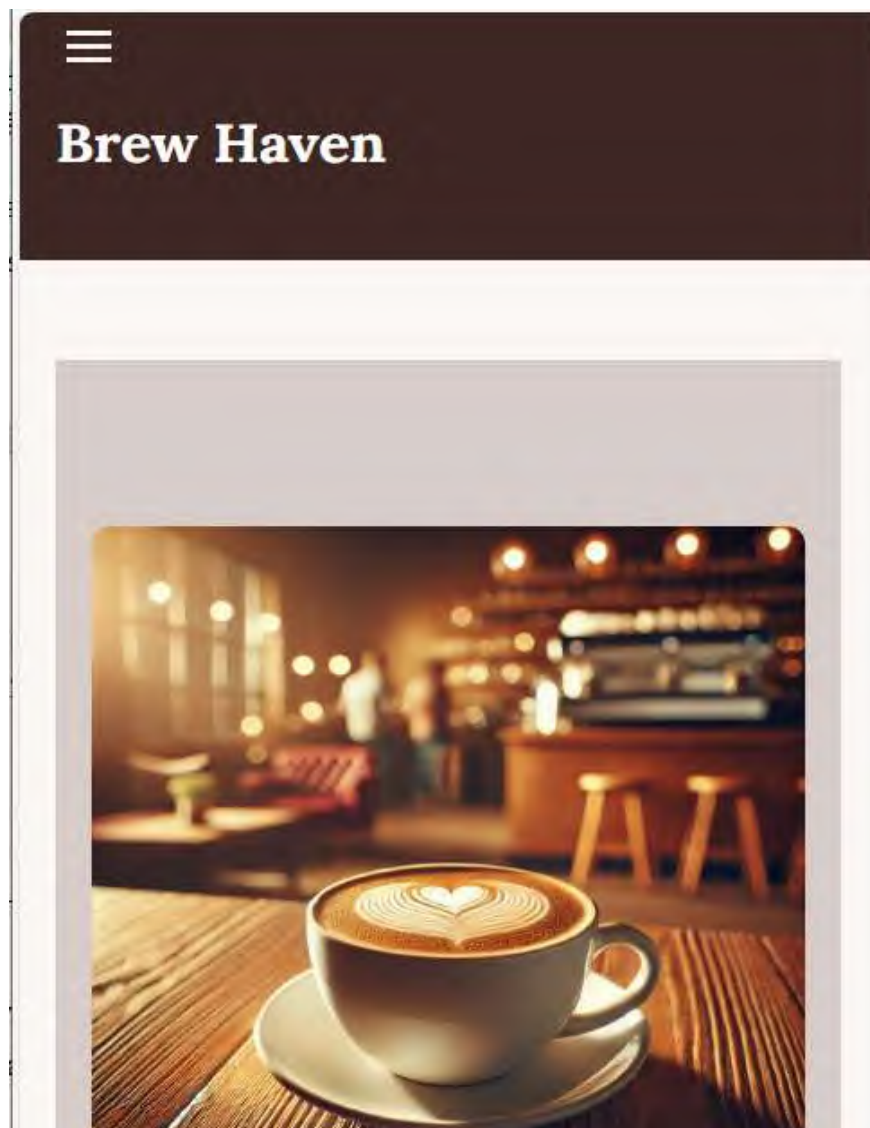


Рисунок 3.21 – Вигляд згорнутого гамбургер-меню при звуженні екрану

Таким чином, було покращено мобільну адаптацію навігації. Зображення сайту в адаптованому вигляді при переході на екран смартфона показано в додатку Г. В подальшому, для розвитку проєкту можна буде додати ефект анімації (перетворення трьох рисочок на хрестик). Також в подальшому варто удосконалити обробку даних з форми, вставити інтерактивну мапу [40].

У результаті виконання роботи вдалося розробити шаблон вебсайту молодіжної кав'ярні, який відповідає усім заданим специфікаціям і досягнув цілей проєкту, використовуючи такі інструменти та ресурси, як Figma та Visual Studio Code.

Опис інших технічних рішень, які були застосовані при розробленні й верстанні вебсайту доцільно провести з демонстрацією отриманих функцій сайту.

3.3 Економічний аналіз варіантів розробки вебсайту

У ході виконання кваліфікаційної роботи розроблено адаптивний шаблон вебсайту молодіжного кафе у формі лендінгу, який може бути використаний як основа для створення вебресурсів закладів ресторанного бізнесу різного формату. Здійснимо економічну оцінку вартості такої розробки для замовника та економічний ефект, а також усереднений підрахунок затрат розробника.

Економічний аналіз розробки в ІТ – це процес кількісного та якісного оцінювання витрат, вигод, ризиків і ефективності, пов'язаних із створенням, впровадженням та експлуатацією інформаційних систем і програмного забезпечення. Метою такого аналізу є прийняття обґрунтованих управлінських рішень щодо доцільності інвестування в ІТ-проєкти, їхньої рентабельності та економічної доцільності [41-42].

Економічний аналіз вартості розробки та ефективності впровадження промо-сайту кав'ярні стане у пригоді для подання інвесторам, обґрунтування витрат або просто оцінки рентабельності. Методи аналізу показано в табл. 3.1.

Таблиця 3.1 – Ключові методи для економічного аналізу

Метод	Для чого використовується
Витратний (Cost-based)	Для оцінки реальної вартості створення сайту
Метод рентабельності (ROI)	Щоб визначити прибутковість сайту
SWOT-аналіз	Оцінка сильних/слабких сторін та можливостей
Метод окупності (Payback Period)	Щоб оцінити, коли інвестиції у сайт себе окуплять
Метод економії витрат (Cost Saving)	Якщо сайт замінює інші канали реклами або обслуговування

Для економічного аналізу розробки сайту використаємо витратний (Cost-based) метод та метод окупності.

Для визначення вартості розробки вебсайту спочатку проведемо розрахунок трудомісткості. Враховуємо, що були використані технології HTML&CSS JavaScript, Visual Studio Code, Figma. Роботу виконував один frontend-розробник, графічні елементи та дизайн підготував графічний дизайнер, тестування здійснив тестуваотник. В розрахунках враховуємо:

- роботу фронтенд-розробника (погодинні ставки);
- графічного дизайнера;
- використання інструментів (VS Code, Figma - безкоштовні/умовно-безкоштовні);
- час виконання;
- тестування адаптивності та кросбраузерності вебсайту [42].

Самим важливим пунктом при розробці продукту є розрахунок його собівартості й визначення кінцевої ціни, оскільки саме вона має основну вагу при визначенні переваг над конкурентами та доцільності розробки системи.

Для розрахунків затрат на оплату праці візьмемо за основу, що розробник працює по стандартному графіку: восьмигодинний робочий день, п'ять днів на тиждень. Оплата праці – погодинна. Вартість години роботи може бути різною в залежності від наступних факторів:

- область застосування системи;
- обсяг застосовуваних інструментів та методів;
- термін виконання;

– складність завдання.

Оплату праці frontend-розробника приймаємо в середньому 350 грн/год [43]. На розробку системи в нашому проєкті виділено 2 тижня (10 робочих днів). Витрати праці графічного дизайнера базового рівня 300 грн/год, працював 16год, включно з усіма фото.

Раніше було встановлено, що вебдизайнер буде працювати по 8 годинному графіку з оплатою 350 грн/год. Загальна формула розрахунку витрат оплати праці:

$$Z_n = (K_{\text{дн}} \times B_{\text{роб}}) \times T_{\text{год}}, \quad (3.1)$$

де Z_n – затрати на оплату праці спеціалістів;

$K_{\text{дн}}$ – кількість робочих днів у проєкті;

$B_{\text{роб}}$ – тривалість робочого дня в годинах;

$T_{\text{год}}$ – погодинна ставка оплати праці розробника.

Підрахунок оплати праці вебдизайнера становить $Z_n = (10 \times 8) \times 350 = 28000$ грн. Оплата праці графічного дизайнера становитиме $Z_{\text{пд}} = (2 \times 8) \times 300 = 4800$ грн. Загальний підрахунок витрат представлено в табл. 3.2.

Таблиця 3.2 – Розрахунок прямих витрат на розроблення вебсайту

№ з/п	Види витрат на розробку і впровадження технології	Витрати часу, одиниць	Грошовий еквівалент робіт, грн/од. часу	Сумарна вартість, грн
1	Розроблення макету вебсайту в програмі Figma (ПЗ безкоштовно) і фото	16 год.	300	4800
2	Розроблення сайту (програміст)	80 год.	350	28000
3	Оплата домену (хостинг) від компанії TheHost, пакет «Діловий»	1 рік	799	799
4	Вартість домену (Міжнародний інформаційний домен Info) на рік	1 рік	249	249
5	Тестування вебсайту (адаптивність, кросбарузерність)	8 год	300	2400
6	Сумарна вартість індивідуального замовлення вебсайту	=(п.1+п.2+ п.3+ п.4+ п.5)		36248

Дані, які використані для підрахунку витрат на розроблення вебсайту, було взято з відкритих джерел. Зокрема, актуальні дані про хостинг взято із вебсайту компанії Thehost [44]. Відомості про середні розцінки роботи вебдизайнера, програміста публікує Upwork [43].

Таким чином, загальна вартість розробки вебсайту одноосібним графічним дизайнером (макет сайту) і вебпрограмістом складе близько 36248 грн станом на початок 2026 р.

Проведемо додатково розрахунок окупності розробленого шаблону вебсайту. Припустимо, що завдяки вебсайту заклад щомісяця залучатиме додатково 10 нових клієнтів. За середнього чека 250 грн додатковий місячний дохід D становитиме: $D = 10 \times 250 \times 30 = 75000$ грн/міс. Навіть, якщо лише 10-15 % цього приросту (7500 – 11250 грн) буде пов'язано із функціонуванням вебсайту, то витрати на його створення можуть окупитися за 4-5 місяців.

Альтернативні підрахунки вартості використання готового шаблону вебсайту для бізнесу не проводилися, оскільки був зроблений вибір з урахуванням маркетингових та інших вигод саме авторської розробки. Підрахунки, наведені в цьому розділі мають оціночний характер. Остаточне рішення приймає кожна компанія залежно від цілей, технічних та інших вимог.

ВИСНОВКИ

У кваліфікаційній роботі проведено аналіз теоретичних основ створення вебсайтів та особливостей їх використання в сучасному бізнес-середовищі, вирішене актуальне завдання розроблення адаптивного шаблону вебсайту для молодіжного кафе у формі лендінгу з використанням сучасних технологій веброзробки та засобів проєктування інтерфейсів. За результатами роботи можна сформулювати такі висновки.

1. Встановлено, що вебсайти залишаються одним із найефективніших інструментів цифрового маркетингу, формування бренду та комунікації з цільовою аудиторією. Особливу увагу приділено ролі лендінгів як інструменту швидкої презентації товарів і послуг та стимулювання користувачів до виконання цільових дій.

2. Досліджено сучасні базові технології веброзробки, а також актуальні підходи до побудови користувацьких інтерфейсів. Окремо розглянуто сучасні тенденції вебдизайну, принципи UI/UX-проєктування та особливості формування позитивного користувацького досвіду.

3. На основі проведеного аналізу сформовано концепцію. Обґрунтовано вибір формату лендінгу як найбільш доцільного рішення для шаблону вебсайту для молодіжного кафе, оскільки він забезпечує компактне представлення інформації, високу швидкість навігації, ефективне використання візуального контенту та зручність перегляду з мобільних пристроїв.

4. Створений дизайн базується на принципах мінімалізму та Modern UI, поєднує сучасні візуальні рішення, адаптивність та орієнтацію на молодіжну аудиторію. Передбачено використання англомовного інтерфейсу, що підвищує універсальність розробленого шаблону та розширює можливості його застосування для міжнародних проєктів ресторанного бізнесу.

5. Для програмної реалізації вебсайту обрано класичний стек фронтенд-розробки, що включає HTML5, CSS3 та JavaScript. Такий підхід забезпечує

незалежність від сторонніх платформ і конструкторів сайтів, високу продуктивність, гнучкість подальшої модернізації.

6. Проведено економічне обґрунтування розробки. Виконано оцінку трудомісткості проєкту, визначено основні статті витрат та розраховано орієнтовну вартість створення вебсайту. Результати аналізу підтвердили економічну доцільність використання розробленого шаблону, швидку окупність.

Практичним результатом роботи став готовий шаблон адаптивного вебсайту молодіжного кафе у формі лендінгу, який містить основні інформаційні блоки: головний екран, розділ про заклад, історію бренду, меню, галерею, відгуки клієнтів, форму замовлення та контактну інформацію. Розроблений шаблон може бути використаний як основа для створення вебресурсів закладів ресторанного бізнесу різного формату з мінімальними витратами часу на адаптацію. Робота пройшла апробацію шляхом оприлюднення тез доповіді на науковій студентській конференції (додаток Д).

В подальшому варто удосконалити обробку даних з форми, вставити інтерактивну мапу, зробити двомовний інтерфейс. Окрім того, можна в автоматизувати прийом замовлень і доставку онлайн. Для цього можемо використовувати різні платіжні системи, такі як PayPal або Stripe, щоб користувачі могли швидко і легко здійснювати покупки на такому сайті. У перспективі можливо додати й AI-асистента для роботи з клієнтами або формування індивідуальної рецептури напоїв.

Розроблений вебсайт може бути корисним для використання і ведення різних видів ресторанного бізнесу невеликими компаніями, розміщення на відповідних платформах-репозитаріях шаблонів вебсайтів, демонстрації прикладного застосування окремих інструментів розробника в навчальному процесі, а також як елемент особистого портфоліо розробника.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Перший у світі вебсайт URL: <https://babel.ua/texts/23307-28-rokiv-tomu-zapustili-pershiy-u-sviti-veb-sayt-mi-pokazali-yak-viglyadav-bi-thebabel-v-1990-roci> (дата звернення 03.05.2026).
2. Історія розвитку вебсайтів. URL: <https://webcraft.by/blog/istoria-razvitia-saitov.html> (дата звернення 03.05.2026).
3. Вебсайт: визначення й застосування URL: <http://www.webtec.com.ua/uk/articles/index/view/2011-05-05/web-site> (дата звернення 03.05.2026).
4. About W3C. W3C: *вебсайт*. URL: <https://www.w3.org/Consortium/> (дата звернення 03.05.2026).
5. Stanadrts. W3C: *вебсайт*. URL: <https://www.w3.org/standards/> (дата звернення 03.05.2025).
6. Вебсторінк. URL: <http://www.znannya.org/?view=concept:364> (дата звернення 03.05.2026).
7. Що таке доменне ім'я? URL: <https://redo.ua/faq> (дата звернення 03.05.2026).
8. How Many Websites Are There in the World? *Siteefy*: вебсайт. URL: <https://siteefy.com/how-many-websites-are-there/#How-Many-Websites-Are-There> (дата звернення 03.05.2026).
9. Бабінський Б. Дослідження форм і статистики інтернет-присутності глобального ринку кав'ярень. Сучасні інформаційні технології та *інноваційні методики в економіці, менеджменті та бізнесі*: Матеріали XXI щорічної студ. наукової конференції, м. Полтава, ПДАУ, 22 квітня 2026 р. Полтава: ПДАУ, 2026 р. С. 80-82.
10. Total number of Websites. URL: https://www.internetlivestats.com/total-number-of-websites/#google_vignette (дата звернення 03.05.2026).
11. Сучасні тенденції застосовування інтернет-технологій URL: https://mmi.fem.sumdu.edu.ua/sites/default/files/mmi2011_4_2_64_74.pdf (дата звернення 03.05.2026).

12. Адаптивний дизайн: Основні принципи та переваги для різних пристроїв URL: <https://freshtech.global/ua/blog/responsive-design-basic-principles-and-benefits-for-different-devices> (дата звернення 03.05.2026).
13. Leosvit Marketing. URL: <https://leosvit.com/> (дата звернення 03.05.2026).
14. Користувацькі сценарії: що це, як і навіщо їх потрібно будувати. *Сценарії взаємодії користувача з промо-сайтом* вебсайт URL : <https://laba.ua/blog/2651-polzovatelskie-scenarii-cto-eto-kak-i-dlya-chego-ih-nuzhno-stroit> (дата звернення 03.05.2026).
15. Поняття, структура та різновиди вебсайтів. URL: <http://www.ndu.edu.ua/liceum/web.pdf> (дата звернення 03.05.2026).
16. Hakon Wium Lie, Bert Bos. Cascading Style Sheets: Designing for the Web. 3rd Edition. R.R. Doneley and Sons Company in Crowfordswill, Indiana, US. 2005. 416 p. ISBN-13: 978-0-321-19312-4.
17. CSS фреймоврки URL: <https://html-plus.in.ua/css-frameworks/> (дата звернення 03.05.2026).
18. URL: <https://w3techs.com/technologies/details/cp-javascript> (дата звернення 03.05.2026).
19. Usage statistics of JavaScript libraries for websites. W3Techs: *вебсайт*. URL: https://w3techs.com/technologies/overview/javascript_library (дата звернення 03.05.2026).
20. 10 JavaScript фреймворків URL: <https://uaspectr.com/2021/03/30/10-javascript-frejmworkiv-2021/> (дата звернення 03.05.2026).
21. JavaScript фреймоврки. URL: <https://www.reclamare.ua/blog/javascript-frejmworki/> (дата звернення 03.05.2026).
22. Бекенд сайту URL: <https://kitweb.pro/bekend-sayta/> (дата звернення 03.05.2026).
23. Що таке UI/UX дизайн. URL: <https://te.itstep.org/blog/ui-and-ux-design> (дата звернення 03.05.2026).

24. Mobile-first підхід і адаптивний дизайн: що нового. URL: <https://atlant.digital/blog/mobile-first-responsive-design-updates> (дата звернення 03.05.2026).
25. 25 трендів вебдизайну у 2025 році: від необруталізму до динамічних інтерфейсів. URL: <https://blog.depositphotos.com/ua/trendi-vebdizajnu-2025.html> (дата звернення 03.05.2026).
26. Вебдизайн. Класичний стиль URL: <https://webstudio2u.net/ua/design-web/417-classic-webdesign.html> (дата звернення 03.05.2026).
27. Мінімалізм у вебдизайні. URL: <http://canis-club.kharkov.ua/minimalizm-u-veb-dizajni/> (дата звернення 03.05.2026).
28. The leading information platform for the global coffee industry. URL: <https://www.worldcoffeeportal.com/about/> (дата звернення 15.04.2026).
29. Які задачі закривають конструктори сайтів? URL: <https://www.worldcoffeeportal.com/about/> (дата звернення 03.05.2026).
30. Промо-сайти для бізнесу. Промо-сайт та що він собою являє? URL: <https://avada-media.ua/ua/services/promo-sajty/> (дата звернення 03.05.2026).
31. 6 Restaurant Landing Page Examples to Get More Guests. *ConvertFlow - Funnel Builder*. URL: <https://www.convertflow.com/campaigns/restaurant-landing-pages> (дата звернення 03.05.2026).
32. Бізнес-план кав'ярні: приклад із розрахунками у 2024 році. *Smart Cafe*. URL: <https://smartcafe.com.ua/infocentr/biznes-plan-kavyarni-priklad-iz-rozrahunkami> (дата звернення 03.05.2026).
33. Ankit K., Sharma V. A Study on User Interface and User Experience Designs and its Tools. *World Journal of Research and Review*. 2021, 43 p.
34. Професійні шаблони вебсайтів для кав'ярень. *Site123*. URL: <https://ua.site123.com/templates/coffee-website-templates> (дата звернення 03.05.2026).
35. What is HTML? *W3C: вебсайт*. URL: <https://www.w3.org/standards/webdesign/htmlcss> ((дата звернення 03.05.2026).

36. Що таке CSS? URL: https://css.in.ua/article/shcho-take-html_10/ (дата звернення 03.05.2026).
37. *Figma*: вебсайт. URL: <https://www.figma.com/design/> (дата звернення 03.05.2026).
38. Visual Studio Code. URL: <https://code.visualstudio.com/> (дата звернення 03.05.2026).
39. State of JS. URL: <https://2018.stateofjs.com/other-tools/> (дата звернення: 23.04.2026).
40. O. Kopishynska, Y. Utkin, I. Sliusar, V. Pysarenko, O. Galych, L. Flehantov, I. Zahrebelna, S. Pysarenko. Integrating Large Language Models into Web Design Study: AI-Assisted Code Optimization in Higher Education / *Proceedings of the 29th World Multi-Conference on Systemics, Cybernetics and Informatics: WMSCI 2025*. 2025. P. 497–504. DOI: <https://doi.org/10.54808/WMSCI2025.01.497>.
41. Laudon K.C., Laudon J.P. *Management Information Systems: Managing the Digital Firm* (16th ed.). Pearson, 2020.
42. Дорош Н.Г. *Економічна ефективність інформаційних технологій*. // Вісник економіки транспорту і промисловості, № 65, 2019.
43. Оцінка економічної ефективності Інтернет-ресурсів: вебсайт. URL: http://dipplus.com.ua/metodicheskiye-ukazaniya-i-informatsiya/article_post/otsinka-yekonomichnoi-yefektivnosti-internet-resursiv (дата звернення 03.05.2026).
44. Ціни годин креативних фахівців та проєктних робіт. URL: <https://cases.media/article/cini-godin-kreativnikh-fakhivciv-ta-proyektnikh-robit-u-2024-roci?srsltid=AfmBOoqXqVXKOD595XObdT3yCXrHsbO4Vu-FFEsQUfPU3u5u7CKEQsNm> (дата звернення 03.05.2026).
45. Тарифні плани. TheHost: Хостинг провайдер. URL: <https://thehost.ua/ua/hosting/plans> (дата звернення 03.05.2026).