

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАОЧНОЇ
ТА ПІСЛЯДИПЛОМНОЇ ОСВІТИ**

КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

другого (магістерського) рівня вищої освіти

на тему: **“Використання методів спрощення розробки веб-сайту із застосуванням РНР-шаблонізатора Twig”**

Виконав: ст. гр. ІТ-71з
Спеціальності 126 – «Інформаційні системи та технології»
(шифр і назва)

Кривецький Юрій Орестович
(Прізвище та ініціали)

Керівник: к.т.н., в.о. доц. Татомир А.В.
(Прізвище та ініціали)

Рецензент: _____
(Прізвище та ініціали)

ДУБЛЯНИ-2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАОЧНОЇ
ТА ПІСЛЯДИПЛОМНОЇ ОСВІТИ

Другий (магістерський) рівень вищої освіти
126 – «Інформаційні системи та технології»

“ЗАТВЕРДЖУЮ”
Завідувач кафедри _____
д.т.н., проф. А.М. Тригуба
“ _____ ” _____ 2023 р.

ЗАВДАННЯ

на кваліфікаційну роботу студенту

Кривецькому Юрію Орестовичу

1. Тема роботи: «Використання методів спрощення розробки веб-сайту із застосуванням PHP-шаблонізатора Twig»

Керівник роботи Татомир Андрій Володимирович, к.т.н., в.о. доцента
Затверджені наказом університету від 17 лютого 2023 року № 331/к-с..

2. Строк подання студентом роботи 15.01.2024 р.

3. Початкові дані до роботи: 1. Вимоги до побудови веб-сайтів.

2. Науково-технічна і довідкова література. 3. Засоби створення та структура веб-сайту. 4. Методика використання PHP-шаблонізаторів.

4. Зміст розрахунково-пояснювальної записки:

1. Аналіз стану питання та веб-сайтів обслуговуючої сфери

2. Порівняння технологій та інструментів створення веб-сайту

3. Використання методів спрощення розробки веб-сайту

4. Імплементация php-шаблонізатора twig

5. Охорона праці та безпека в надзвичайних ситуаціях

Висновки

Бібліографічний список.

Додатки

5. Перелік презентаційного матеріалу : 1 та 2 – Тема, мета, завдання роботи;
3 – Аналіз стану галузі інформаційних послуг; 4 – Головні поняття та класифікація веб-сайтів; 5 – Огляд сайтів аналогів; 6 – Проектування web-сторінки; 7 – Розробка структури веб-сайту; 8 – Специфікація вимог; 9 – Програмна реалізація; 10 та 11 – вибір технологій розробки; 12 та 13 – РНР-шаблонізатор Twig; 14 – тестування; 15 – Висновки.

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 4	<i>Татомир А.В., в.о. доцента кафедри інформаційних технологій</i>		
5	<i>Городецький І.М., доцент кафедри фізики, інженерної механіки та безпеки виробництва</i>		

7. Дата видачі завдання 17 лютого 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Етапи виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	<i>Ознайомлення з темою роботи. Структура. Вступ. Підготовка матеріалів.</i>	17.02 - 01.03.23	
2.	<i>Формулювання задачі створення веб-сайтів. Вивчення основних принципів</i>	01.03 - 01.06.23	
3.	<i>Проектування структури та дизайну сайту. Вибір технологій та інструментів для розробки</i>	01.06 - 01.08.23	
4.	<i>Розробка функціональності та наповнення веб-сайту</i>	01.06 - 01.08.23	
5.	<i>Тестування та усунення помилок. Практичне використання сайту в закладах</i>	01.08 - 01.10.23	
6.	<i>Охорона праці та безпеки в надзвичайних ситуаціях.</i>	01.10 - 01.11.23	
7.	<i>Формування висновку з результатів роботи</i>	01.10 - 01.11.23	
8.	<i>Підготовка бібліографічного списку. Підготовка додатків</i>	01.12 - 01.01.24	
9.	<i>Завершення написання кваліфікаційної роботи.</i>	01.01 - 15.01.24	

Студент _____ Кривецький Ю.О.
 (підпис)

Керівник роботи _____ Татомир А.В.
 (підпис)

Кваліфікаційна робота: 70 с. текст. част., 24 рис., 2 табл., 36 джерел.

Використання методів спрощення розробки веб-сайту із застосуванням PHP-шаблонізатора Twig. Кривецький Ю.О. Кафедра ІТ. – Дубляни, Львівський НУП, 2024.

Проаналізовано залучення користувачів Інтернету в Україні. Наведено головні поняття та класифікацію веб-сайтів. Виконано аналіз специфіки побудови веб-сайтів аналогів.

Виконано огляд інструментів для розробки сайтів. Описано доцільність застосування технологію HTML – мову розмітки гіпертексту, CSS – каскадні таблиці стилів. Наведено порівняльні переваги використання інструментів для автоматизації Gulp. Описано структуру та переваги застосування пакетного менеджера npm, доцільність використання JavaScript.

Описано причини та переваги використання PHP-шаблонізатора Twig в кваліфікаційній роботі.

Наведено використання елементів середовища розробки Visual Studio Code. Описано розробку структури веб-сайту, етапи створення макету, реалізацію функціональності веб-сайту, верстку та оформлення, тестування та оптимізації сайту, розгортання та підтримки.

Описано інструкцію відвідувача веб-сайту.

Запропоновано заходи щодо охорони праці та безпеки в надзвичайних ситуаціях.

Ключові слова: веб-сайт, розробка, PHP-шаблонізатор, Twig, онлайн-замовлення, сервіс, ефективність.

ЗМІСТ

РОЗДІЛ 1. АНАЛІЗ СТАНУ ПИТАННЯ ТА ВЕБ-САЙТІВ ОБСЛУГОВУЮЧОЇ СФЕРИ.....	9
1.1 Аналіз залучення користувачів Інтернету в Україні.....	9
1.2 Головні поняття та класифікація веб-сайтів	11
1.3 Аналіз специфіки побудови веб-сайтів аналогів	14
РОЗДІЛ 2. ПОРІВНЯННЯ ТЕХНОЛОГІЙ ТА ІНСТРУМЕНТІВ СТВОРЕННЯ ВЕБ-САЙТУ	19
2.1 Огляд інструментів для розробки сайту	19
2.2. Технологія HTML – мова розмітки гіпертексту	20
2.3. Технологія CSS – каскадні таблиці стилів	21
2.4 Використання інструментів для автоматизації Gulp.....	24
2.5 Пакетний менеджер npm	26
2.6 Мова програмування JavaScript.....	28
2.7 PHP-шаблонізатор Twig	31
РОЗДІЛ 3. ВИКОРИСТАННЯ МЕТОДІВ СПРОЩЕННЯ РОЗРОБКИ ВЕБ-САЙТУ	34
3.1 Середовище розробки Visual Studio Code	34
3.2 Розробка структури веб-сайту	35
3.2 Етапи створення макету веб-сайту.....	38
3.3 Реалізація функціональності веб-сайту	39
3.4 Верстка та оформлення	40
3.5 Тестування та оптимізація сайту.....	50
3.6 Розгортання та підтримка	51
РОЗДІЛ 4. ІМПЛЕМЕНТАЦІЯ PHP-ШАБЛОНІЗАТОРА TWIG	54
4.1 Інструкція відвідувача.....	54
РОЗДІЛ 5. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	58
5.1. Розробка логіко-імітаційної моделі виникнення травм і аварій	58
5.2. Планування заходів із покращення умов праці	60
5.3. Безпека в надзвичайних ситуаціях	61
ВИСНОВКИ	62

	6
БІБЛІОГРАФІЧНИЙ СПИСОК.....	64
Додаток А.....	68
Фрагмент коду сторінки меню.....	68

ВСТУП

Ресторанна індустрія наразі є однією з найперспективніших галузей, що розвиваються. Успіх будь-якого бізнесу, в тому числі й ресторанного, залежить від різних факторів. Управління рестораном – це складний процес, який включає в себе контроль бухгалтерських процедур, моніторинг поведінки персоналу, аналіз транзакцій, відстеження запасів, визначення собівартості страв та інгредієнтів, управління простроченими продуктами, забезпечення дотримання санітарних і технологічних норм. Автоматизація цих процесів має вирішальне значення через велику кількість деталей.

Переваги автоматизації операцій у всіх закладах харчування очевидні не тільки для власників, а й для клієнтів. Автоматизація дозволяє пришвидшити оплату рахунків, запровадити ефективні системи знижок для зареєстрованих клієнтів, оптимізувати управління чергою та забезпечити наявність меню з усіма необхідними інгредієнтами. У сучасну цифрову епоху реклама в Інтернеті стає все більш актуальною, оскільки кількість користувачів Інтернету продовжує зростати. Добре організований і привабливий веб-сайт може значно збільшити кількість відвідувачів ресторану.

Інтернет-реклама є економічно вигідною і дає можливість оперативно оновлювати або змінювати інформацію. Вона також дозволяє отримувати відгуки від клієнтів та проводити опитування, що дає змогу ресторану покращити якість їжі, обслуговування та загальний клієнтський досвід. Важливо відзначити, що надійний сайт з меню слугує цінним інструментом для підтримки позитивного іміджу закладу.

Мета роботи – розробка сайту з меню закладу для автоматизації роботи ресторану.

Об'єкт роботи – процес взаємодії відвідувачів ресторану зі сайтом, технологія розробки із PHP-шаблонізатором Twig.

Предмет роботи: показники відвідуваності сайту та продажу замовлень, функціонал для перегляду меню, замовлення страв та іншої взаємодії з

рестораном через Інтернет.

Завдання роботи:

- 1 проаналізувати інформаційні сервіси для підтримки ведення ресторанного бізнесу;
- 2 розглянути програмні рішення для роботи сайтів-ресторанів, оцінити їх функціональні можливості та придатність для використання;
- 3 сформулювати вимоги до сайту, визначити необхідний функціонал, архітектуру системи та інші технічні деталі;
- 4 створити детальний план розробки системи, розробити веб-сайт із використанням PHP-шаблонізатора TWIG, включаючи вибір технологій;
- 5 розробити інструкцію відвідувача.

Новизна роботи:

Виконано порівняльну оцінку програмних рішень щодо роботи сайтів із надання послуг виготовлення та доставки їжі.

Описано вимоги до сайту із доцільністю використання відповідного функціоналу, архітектуру системи та інші технічні деталі.

Створено план розробки веб-сайту, реалізовано цей план із використанням PHP-шаблонізатора TWIG, включаючи вибір технологій.

РОЗДІЛ 1. АНАЛІЗ СТАНУ ПИТАННЯ ТА ВЕБ-САЙТІВ ОБСЛУГОВУЮЧОЇ СФЕРИ

1.1 Аналіз залучення користувачів Інтернету в Україні

Історія українського Інтернету почалася в грудні 1990 року, коли Міжгалузевий науковий центр технології програмування "Технософт", створений Юрієм Янковським, підключився до Всесвітньої мережі в Києві. Через кілька місяців компанія стала першим в Україні інтернет-провайдером.

Сьогодні, до початку повномасштабного військового вторгнення, всесвітньою мережею в Україні користувалися близько 29 млн. українців (рис. 1.1) [3].

СИТУАЦІЯ ДО ПОЧАТКУ ПОВНОМАСШТАБНОГО ВТОРГНЕННЯ

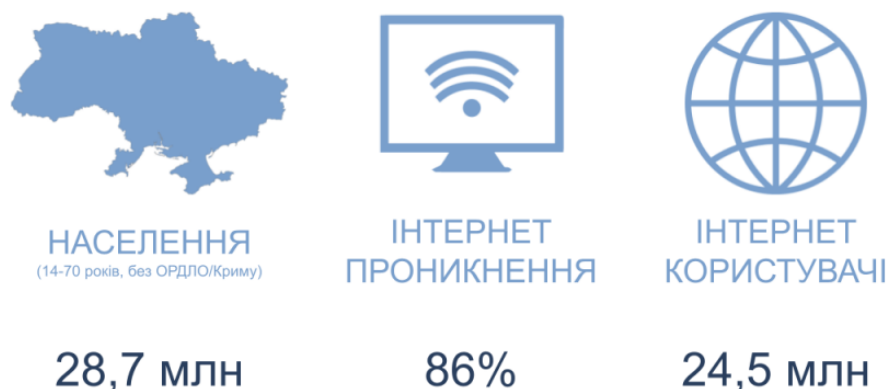


Рисунок 1.1. – Загальна Інтернет-аудиторія України

Водночас, більшість українських користувачів – у віці від 14 до 70 років.

В даних дослідження Інтернет-аудиторії України gemiusAudience з 1 червня 2023 року здійснено переоцінку популяції у зв'язку з війною в Україні. Загальна Інтернет-аудиторія України на всіх платформах в червні 2023 року становить 25,6 млн Real Users (зменшилась на 2,7 млн Real Users). Загальна аудиторія на ПК становить 14,2 млн Real Users (зменшилась на 5,4 млн Real Users), на Телефонах – 23,7 млн Real Users (зменшилась на 1,7 млн Real Users), на Планшетах – 2,9 млн Real Users.

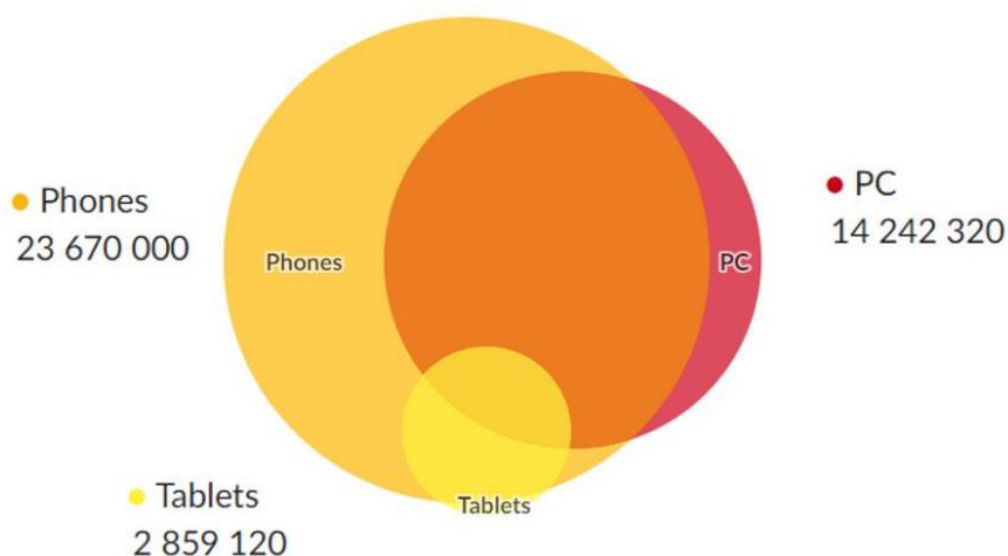


Рисунок 1.2. – Інтернет-аудиторія України (червень 2023 рік)

Чемпіонами за охопленням на платформах ПК+Телефони у червні залишаються всі ті ж гравці діджитал сегменту що й до переоцінки популяції УАнету. На першому місці – сервіс Google (88,5%), на другому – соціальна мережа FB (54,4%) і вже на третій позиції – YouTube (52,5%), потім стандартно вільна онлайн бібліотека Wikipedia (40,1%) з відчутним відривом в охопленні.

На п'ятій та шостій позиціях розмістились новинні сайти – UNIAN (39,1%) та TSN (37,3%). Наступні у рейтингу – маркетплейси Rozetka (36,6%) та PROM (34,4%) та закривають топ-10 – портал UKR.net (31%), банк Privatbank (29,7%).

ТОП 20 САЙТІВ ЗА ОХОПЛЕННЯМ			ЧЕРВЕНЬ 2023	
	DOMAIN	REAL USERS	REACH	AverageTimeSpent
1	google.com	22 691 880	88.50%	4h 6m 2s
2	facebook.com	13 947 480	54.40%	52m 21s
3	youtube.com	13 457 880	52.49%	41m 55s
4	wikipedia.org	10 289 160	40.13%	7m 15s
5	unian.net	10 013 760	39.05%	14m 8s
6	tsn.ua	9 562 320	37.29%	18m 3s
7	rozetka.com.ua	9 389 160	36.62%	8m 54s
8	prom.ua	8 818 920	34.39%	12m 50s
9	ukr.net	8 693 640	33.91%	51m 5s
10	privatbank.ua	7 937 280	30.96%	36m 2s
11	instagram.com	7 612 200	29.69%	12m 39s
12	olx.ua	7 314 120	28.53%	54m 45s

13	24tv.ua	7 220 520	28.16%	8m 34s
14	sinoptik.ua	6 866 640	26.78%	3m 29s
15	tiktok.com	6 703 560	26.14%	3m 45s
16	RBC-Ukraine	6 397 200	24.95%	17m 3s
17	nv.ua	5 932 800	23.14%	7m 37s
18	epicentrk.ua	5 847 120	22.80%	5m 38s
19	pravda.com.ua	5 812 200	22.67%	27m 31s
20	glavred.info	5 723 640	22.32%	7m 8s

Найдовший середній проведений час на сайті незмінно у Google, а за ним маркетплейс OLX, який повернув свою другу позицію та соціальна мережа FB.

1.2 Головні поняття та класифікація веб-сайтів

Веб-сторінка – це інформаційний ресурс доступний в мережі World Wide Web (Всесвітня павутина), який можна переглянути у веб-браузері зазвичай, ця інформація записана в форматі HTML або XHTML, і може містити гіпертекст з навігаційними гіперпосиланнями на інші веб-сторінки.

Гіпертекст – це текст для перегляду на комп'ютері, який містить зв'язки з іншими документами («гіперзв'язки» чи «гіперпосилання»); читач має змогу перейти до пов'язаних документів безпосередньо з вихідного (первинного) тексту, активізувавши посилання. Загальновідомим та яскраво вираженим прикладом сучасного гіпертексту слугують веб-сторінки, підготовлені за допомогою HTML (мова розмітки гіпертексту) й розміщені у web-мережі. Відповідно, в комп'ютерній термінології під гіпертекстом розуміється текст, сформований за допомогою мови розмітки, що потенційно містить у собі гіперпосилання. Гіперпосилання дозволяють переходити від вихідного (одного) тексту до безлічі інших текстів, розміщених у web-мережі.

Елементи веб-сторінок:

Контент – змістове наповнення веб-сторінки, доступне користувачеві:

1) за формою подачі: текст, зображення, аудіо, відео;

2) за доступністю: безкоштовний, платний;

3) за унікальністю: унікальний, неунікальний.

Елементи навігації – засоби для переходу на інші веб-сторінки.

Елементи дизайну – елементи структурування контенту та його форматування, оформлення сторінки.

Додаткові елементи веб-сторінок:

Банер – це невеличке рекламне зображення, що розміщується на одній сторінці, а посилається на іншу.

Форма – частина веб-сторінки, до якої користувач може записувати свої відомості та надсилати їх на сервер, де розміщується веб-сторінка.

Веб-сайт – сукупність веб-сторінок, доступних у Інтернеті, які об'єднані як за змістом, так і навігаційно.

Сайтом також називають вузол мережі Інтернет, комп'ютер, за яким закріплена унікальна IP-адреса, і взагалі будь-який об'єкт в Інтернеті, за яким закріплена адреса, що ідентифікує його в мережі (FTP-site, WWW-site тощо).

Структура сайту

- Внутрішня – дозволяє всі необхідні дані розмістити зручно і доступно в розділах та підрозділах.

- Зовнішня – визначає спосіб розміщення відомостей на сторінці при виборі кожного пункту меню, визначає дизайн сайту.

Внутрішня структура сайту

- Лінійна (послідовна) – доцільно використовувати у разі послідовного подання даних та відомостей; перегляд послідовний: від початкової до останньої сторінки сайту.

- Ієрархічна – одна сторінка головна, яка не має попередніх, решта сторінок мають лише одну попередню сторінку.

Довільна – кожна сторінка може містити посилання на довільну кількість сторінок сайту.

- Гібридна – поєднання ієрархічної і послідовної структур.

- Мережна – базується на побудові системи такої навігації сайту, коли між горизонтальними та вертикальними елементами існує взаємний зв'язок та можливості швидкого переходу з однієї сторінки на іншу без додаткової необхідності відвідування проміжних сторінок.

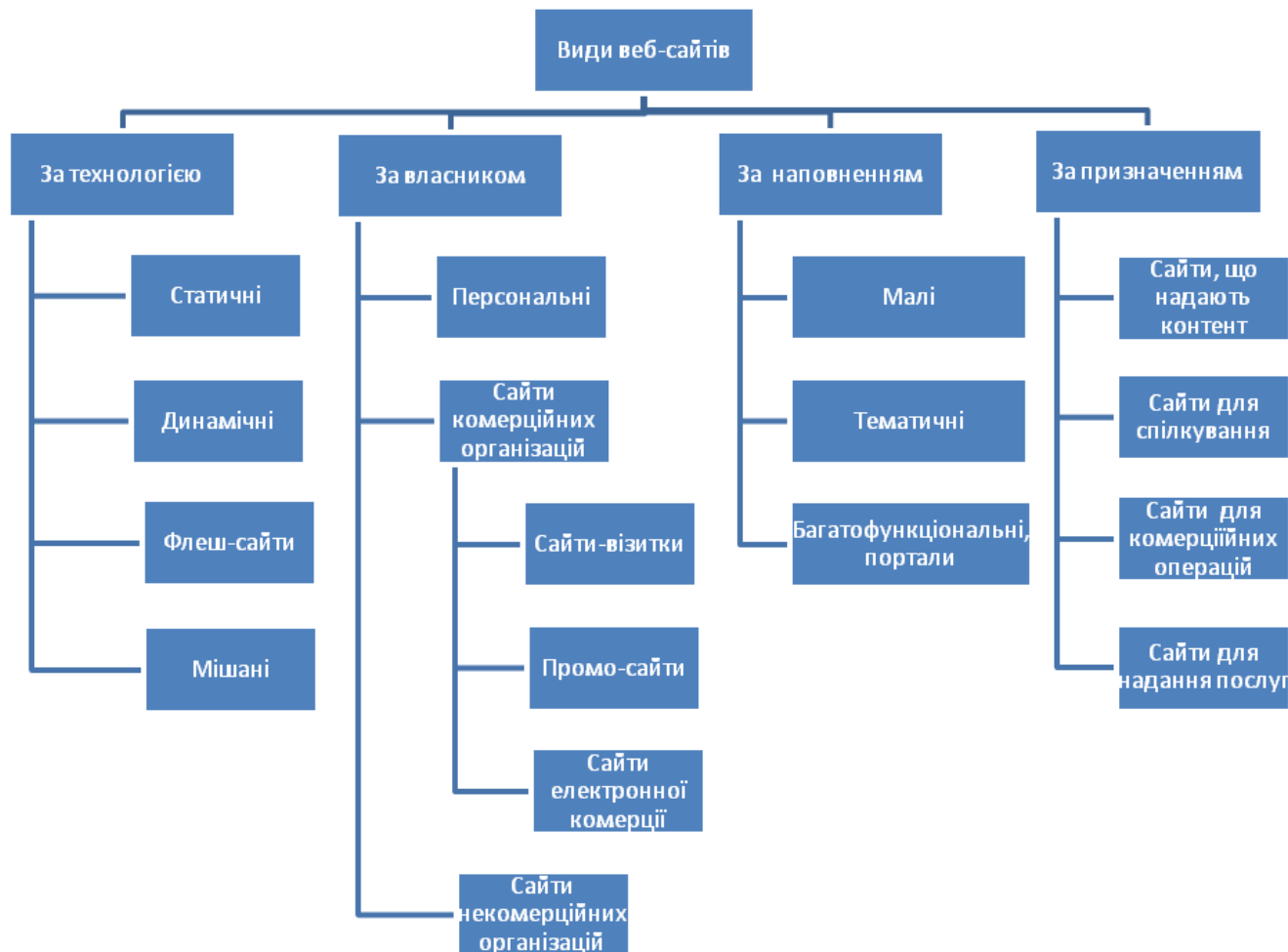


Рисунок 1.3 – Класифікація веб-сторінок

Головна сторінка сайту має задовольняти правило “3 по 3”, тобто 3 абзаци по 3 речення про тематику, зміст, авторів та призначення сайту. Інколи замість головної сторінки з'являється заставка – велике фото зі звуковим супроводом або відеороликом. Такі сторінки створюють, що привернути увагу користувачів.

Основні вимоги до веб-сайтів:

- Наявність головної сторінки зі стислим каталогом усіх розділів (рубрик) сайту. З кожної сторінки сайту мають бути посилання на головну сторінку.
- Наявність швидкої і зручної навігації та пошуку.

- Наявність на головній сторінці сайту стислої інформації для відвідувачів сайту про його структуру і змістове наповнення.

- Оновлення сторінок сайту.

- За наявності великої кількості текстової інформації на сайті доречно розміщувати окрім web-сторінок з текстами також архіви у форматі *Zip* чи *Rar*.

Сторінки сайту мають бути зручними для користувачів сайту, а також для роботи з ними.

- Інформація на сайті може бути представлена українською, російською, англійською мовами.

- Сайт повинен містити засоби зворотнього зв'язку.

Оцінювання веб-сайтів:

- Ступінь інформативності;
- Частота оновлення даних на сайті;
- Наявність інтерактивних засобів;
- Доступні мови для відтворення веб-сайту;
- Візуальна привабливість сайту;
- Зручність користування;
- Зрозумілість для користувача.

Таким чином, багато сайтів на якийсь спосіб займаються заробітком, через розміщення реклами або надання платних послуг користувачам. Однак, вони були створені спочатку не для прибутку, а для задоволення потреб користувачів, тому їх можна вважати некомерційними. Але в таких випадках границя між комерційними і некомерційними сайтами стає дещо нечіткою, і багато з цих сайтів можна назвати "умовно некомерційними".

1.3 Аналіз специфіки побудови веб-сайтів аналогів

Важливим завданням для розуміння нашої розробки є аналіз функціоналу найбільш популярних web-сервісів у сфері створення меню для закладів

харчування. Упершу чергу необхідно описати загальні принципи роботи відповідних сервісів. Для аналізу обрано декілька сайтів зі схожою тематикою і призначенням.

Сайт «Burger King». Сайт є офіційним веб-ресурсом популярного ресторану швидкого харчування "Burger King" в Польщі. Заклад має яскравий і впізнаваний дизайн, з використанням характерних корпоративних кольорів - червоного, жовтого та білого. Дизайн відображає бренд "Burger King" і створює енергійну та дружню атмосферу. Сайт має зручну навігаційну структуру, що дозволяє легко знайти потрібну інформацію. Навігаційне меню розташоване вгорі сторінки та містить розділи, такі як "Меню", "Акції", "Наша історія" та інші. На сайті доступне повне онлайн-меню "Burger King", де представлені різноманітні бургери, снеки, напої та інші страви. Кожна позиція меню супроводжується описом, фотографією та ціною. Також, можливість додати певні опції до замовлення, такі як додаткові інгредієнти або зміна порційного розміру.

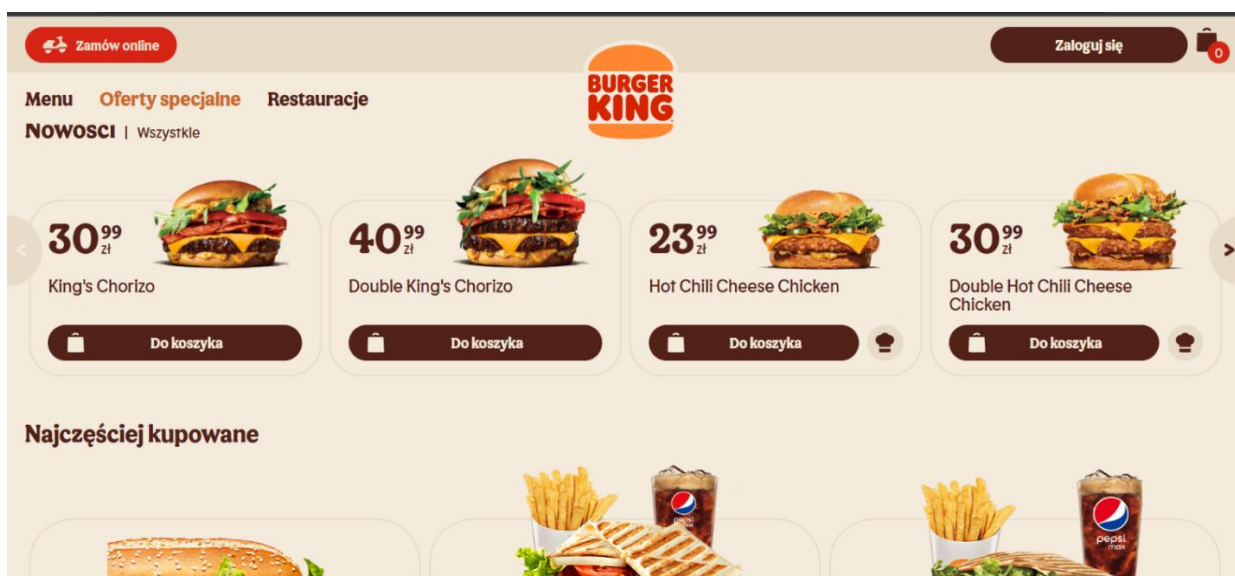


Рисунок 1.4 – Сторінка меню Burger King

Сайт "Burger King" пропонує ряд додаткових функцій для зручності відвідувачів. Серед них можна виділити можливість замовити їжу онлайн з можливістю доставки або самовивозу. Також, є опція зареєструватись на сайті та отримувати спеціальні пропозиції та знижки. На сайті є розділ, присвячений історії компанії "Burger King", де можна дізнатися більше про їхні цінності,

філософію та досягнення.

Загалом, сайт "Burger King" пропонує відвідувачам зручний спосіб ознайомитися з їхнім меню, акціями та послугами. Він надає можливість замовити їжу онлайн і зручно організувати свій візит до ресторану.

Сайт «Meat and Burger» має сучасний та стильний дизайн. Він використовує темну колірну палітру, що створює елегантну та привабливу атмосферу. Дизайн сайту підкреслює концепцію м'яса та бургерів, які є основою ресторану. Сайт має зручну навігаційну структуру, що дозволяє легко знайти інформацію. Головне меню розташоване вгорі сторінки, а на головній сторінці є швидкий доступ до основних розділів, таких як «Меню», «Про нас», «Контакти» та інші. Сайт надає доступ до повного онлайн-меню «Meat and Burger». Тут представлені різноманітні види бургерів, снеки, страви з м'яса та інші страви. Кожне блюдо супроводжується фотографією, описом та ціною.

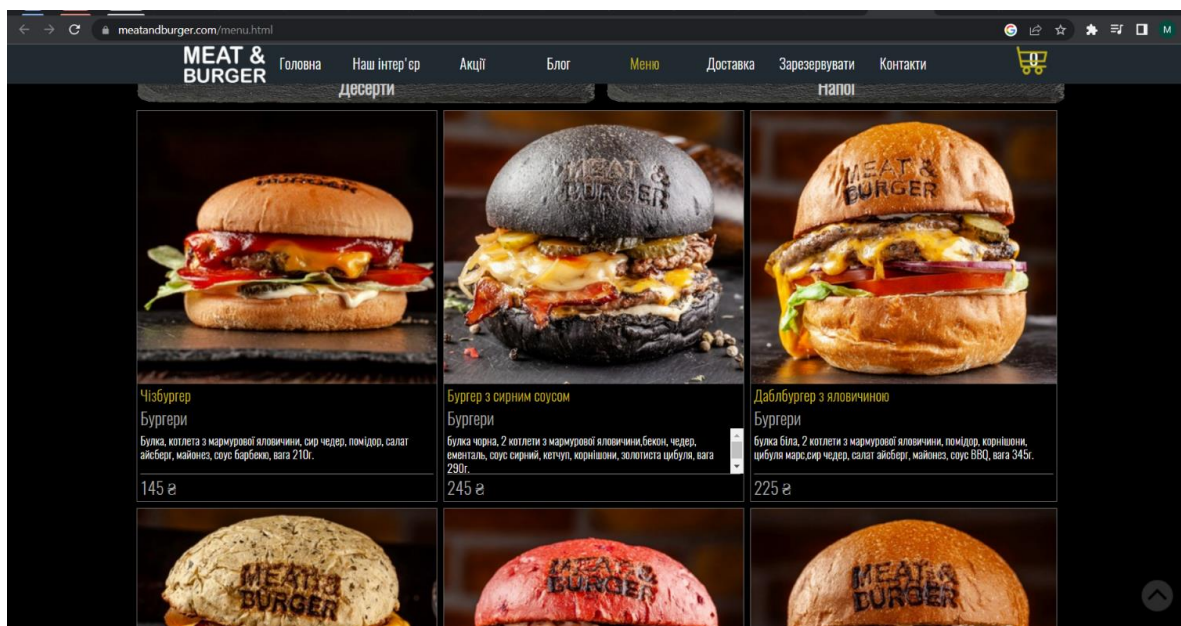


Рисунок 1.5 – Головна сторінка сайту Meat and Burger

Сайт містить інформацію про ресторан «Meat and Burger». Ви можете дізнатися більше про їхню концепцію, якість продуктів, команду та стиль обслуговування. Також є розділ з відгуками клієнтів. На сайті надано контактну інформацію ресторану, включаючи адресу, номер телефону та електронну пошту. Крім того, можна скористатися формою зворотного зв'язку для зручності клієнтів.

Сайт «Meat and Burger» надає можливість ознайомитися з різноманітним меню ресторану, дізнатися більше про його концепцію та зв'язатися з ними для замовлення або отримання додаткової інформації.

Сайт «Pasibus». Сучасний та привабливий дизайн. Його кольорова схема створює енергійну та жваву атмосферу. Дизайн сайту чітко передає концепцію ресторану, яка зосереджена на бургерах. Навігація на сайті здійснюється за допомогою горизонтального меню, яке містить розділи, такі як «Бургери», «Напої», «Акції» та інші. Навігаційні елементи зручно розташовані вгорі сторінки, що полегшує пошук потрібної інформації. Сайт «Pasibus» пропонує повне онлайн-меню з бургерами, їхніми складовими та цінами. Ви можете переглянути різні види бургерів, включаючи вегетаріанські та веганські варіанти. Кожен бургер супроводжується фотографією та описом.

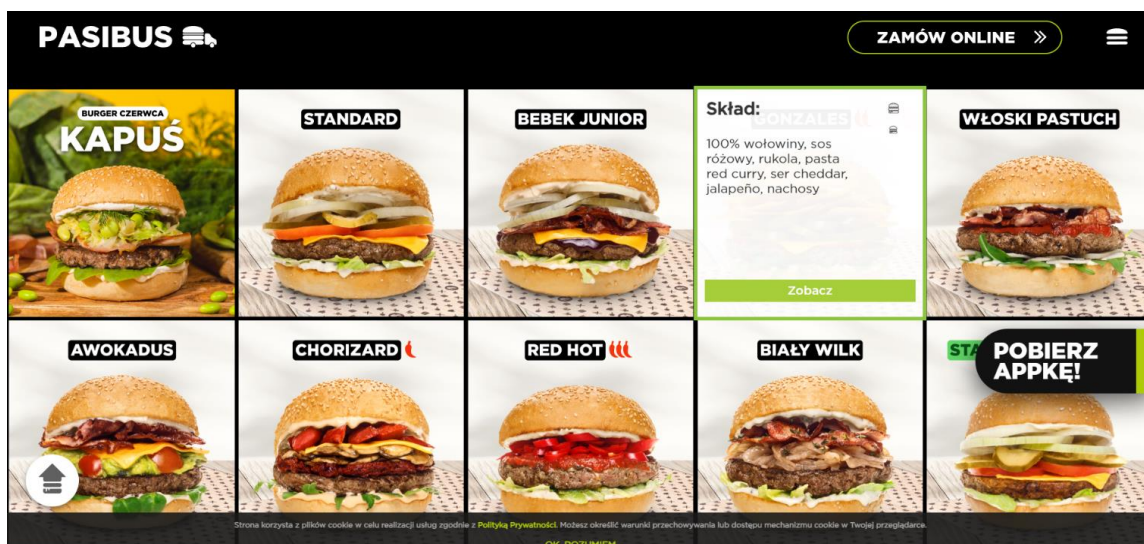


Рисунок 1.6 – Сторінка меню сайту Pasibus

Сайт надає детальну інформацію про ресторан «Pasibus». Ви можете дізнатися більше про їхню філософію, якість продуктів, походження інгредієнтів та підхід до готування. Також є інформація про розділи меню, які не стосуються бургерів, наприклад, напої. Сайт містить контактну інформацію ресторану, включаючи адресу, номер телефону та посилання на соціальні мережі. Також є можливість зв'язатися з рестораном за допомогою онлайн-форми зворотного зв'язку.

Таблиця 1.1 – Порівняння аналогів сайтів

Характеристика	Pasibus	Meat and Burger	Burger King
<i>Функціонал</i>	Онлайн-меню, зручна навігація, контактна інформація, форма зворотного зв'язку	Онлайн-меню, контактна інформація, огляди клієнтів	Онлайн-меню, контактна інформація, огляди клієнтів
<i>Дизайн</i>	Сучасний, жвавий, енергійний	Стильний, сучасний	Корпоративний, чіткий
<i>Схожість тематики</i>	Спеціалізація на бургерах, багато варіантів страв	Спеціалізація на м'ясі та бургерах	Широкий асортимент страв
<i>Зворотній зв'язок</i>	Форма зворотного зв'язку, контактна інформація	Контактна форма зворотного зв'язку, контактна інформація	Контактна форма зворотного зв'язку, контактна інформація
<i>Збільшення фото меню</i>	Не вказано	Присутнє	Присутнє
<i>Цінова політика</i>	Ціни вказані в меню, прозора цінова політика	Ціни вказані в меню	Ціни вказані в меню
<i>Інгредієнти</i>	Інформація про склад бургерів, вегетаріанські та веганські опції	Інформація про склад бургерів	Інформація про склад страв
<i>Зручність використання</i>	Зручна навігація, легко знаходити необхідну інформацію	Зручна навігація, легко знаходити необхідну інформацію	Зручна навігація, легко знаходити необхідну інформацію

Отже, проаналізувавши сайти зі схожою тематикою і призначенням було вирішено розробити сайт який не матиме недоліків виявлених на сайтах представлених в таблиці. Дана розробка матиме всі переваги даних сайтів, а також матиме унікальний дизайн та зручний інтерфейс.

РОЗДІЛ 2. ПОРІВНЯННЯ ТЕХНОЛОГІЙ ТА ІНСТРУМЕНТІВ СТВОРЕННЯ ВЕБ-САЙТУ

2.1 Огляд інструментів для розробки сайту

Один з основних інструментів для розробки веб-сайтів – це текстовий редактор, або інтегроване середовище розробки (IDE). Вони дозволяють розробникам створювати і редагувати веб-код, такий як HTML, CSS і JavaScript. Деякі популярні текстові редактори для розробки веб-сайтів включають Sublime Text, Visual Studio Code, Atom і Notepad++.

Для організації файлів і керування версіями розробникам можуть бути корисні системи керування версіями, такі як Git. Git дозволяє зберігати копії різних версій файлів, спільно працювати над проектом з іншими розробниками та відстежувати зміни, що були внесені в код.

Для стилізації веб-сайту і надання йому зовнішнього вигляду використовуються каскадні таблиці стилів (CSS). Існують різні інструменти для редагування CSS, включаючи редактори, які надають зручний інтерфейс для зміни стилів, і фреймворки, які надають готові набори стилів і компонентів.

Окрім того, для програмування функціональності веб-сайту використовуються мови програмування, такі як JavaScript, PHP, Python та інші. Ці мови мають свої інструменти розробки, включаючи середовища виконання, компілятори та бібліотеки [21].

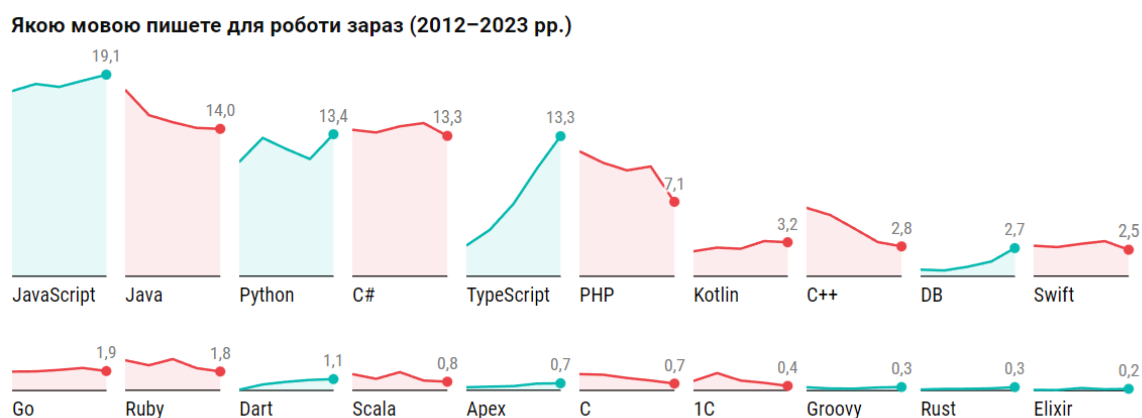


Рисунок 2.1 – Найпопулярніші мови програмування у 2012-2023 роках

В даний час перед web-розробником ставиться безліч різних завдань – від створення інтерактивних розважальних сайтів до серйозних бізнес проектів, які потребують підвищеної надійності і безпеки від несанкціонованого доступу.

Для їх реалізації потрібні правильно підібрані інструменти – мови програмування, фреймворки або, що стають все більш актуальними, системи управління контентом. Сьогодні доступно багато мов програмування.

Перевага кожної із них може бути виражена лише в контексті завдання. Вибір мови залежить від того, які знання має програміст та чи достатньо їх для реалізації проекту [22].

2.2. Технологія HTML – мова розмітки гіпертексту

HTML (Hypertext Markup Language) – це мова розмітки гіпертексту, яка використовується для створення веб-сторінок. Вона визначає структуру та семантику контенту на сторінці, що дозволяє браузерам правильно інтерпретувати та відображати цей контент для користувачів.

Основні поняття HTML:

1 **Теги:** HTML складається з тегів, які визначають різні елементи на сторінці. Кожен тег має відкриваючий та закриваючий тег, які обмежують контент, що міститься в них.

2 **Структура:** HTML використовує структуру з вкладеними елементами. Найвищий рівень структури – елемент `<html>`, який охоплює всі інші елементи сторінки, такі як `<head>` та `<body>`.

3 **Елементи:** HTML має широкий набір елементів для розмітки різних типів контенту. Наприклад, `<h1>`-`<h6>` використовуються для заголовків різних рівнів, `<p>` - для абзаців тексту, `` - для вставки зображень і так далі.

4 **Атрибути:** Елементи HTML можуть мати атрибути, які надають додаткову інформацію про ці елементи. Наприклад, атрибут `src` використовується в елементі `` для вказівки шляху до зображення.

5 **Гіперпосилання:** HTML дозволяє створювати гіперпосилання за допомогою елемента <a>. Це дозволяє створювати взаємозв'язки між різними сторінками та ресурсами в Інтернеті.

6 **Форми:** HTML надає можливість створювати форми для збору даних від користувачів. Елементи, такі як <input>, <select>, <textarea>, використовуються для створення полів введення, вибору та текстових областей.

7 **Семантика:** HTML також пропонує семантичні елементи, які надають додаткову інформацію про зміст сторінки. Наприклад, <header>, <nav>, <main>, <article>, <footer> - це елементи, які вказують на різні частини сторінки.

Загальна структура базового HTML-документа включає наступні елементи:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Назва сторінки</title>
</head>
<body>
  <!-- Зміст сторінки тут -->
</body>
</html>
```

HTML є основою веб-розробки і використовується разом з CSS (Cascading Style Sheets) та JavaScript для створення багатофункціональних веб-додатків. Розуміння HTML є важливим для створення веб-сторінок з належною структурою та семантикою, що сприяє успіху у веб-розробці [3].

2.3. Технологія CSS – каскадні таблиці стилів

CSS (Cascading Style Sheets – каскадні таблиці стилів) – це мова, яка використовується для визначення візуального представлення та форматування веб-сторінок. Вона дозволяє розробникам визначати положення, колір, шрифт, фон та інші властивості HTML-елементів на сторінці.

Ключовою концепцією CSS є її каскадна природа. Це означає, що стилі можна застосовувати на різних рівнях, наприклад, у розмітці сторінки, в окремих файлах стилів або безпосередньо в елементах HTML. Правила пріоритету використовуються для визначення того, які стилі мають перевагу, коли на елементи впливають декілька правил. CSS пропонує широкий спектр властивостей і селекторів, які забезпечують точний контроль над зовнішнім виглядом елементів. За допомогою CSS розробники можуть змінювати розмір, колір, шрифт, межі, фон, позиціонування тощо. Це дозволяє створювати стильні та візуально привабливі веб-сторінки, забезпечуючи при цьому узгодженість на різних пристроях і браузерах [2].

CSS відіграє невід'ємну роль у веб-розробці і використовується разом з HTML та JavaScript для створення динамічних та інтерактивних веб-сторінок. Він дає змогу розробникам керувати візуальними аспектами сторінок, покращувати естетику та користувацький досвід. За допомогою CSS розробники можуть створювати інтерфейси, які відповідають бренду, дизайну та фірмовому стилю веб-сайту або програми. Змінивши один файл стилів, можна легко змінити весь зовнішній вигляд веб-сайту, що спрощує процес внесення оновлень і модифікацій.

Крім того, CSS дозволяє створювати адаптивні та адаптивні веб-дизайни, які можуть без проблем відображатися на різних пристроях та екранах. Медіа-запити та гнучкі властивості CSS дозволяють налаштовувати розміщення та розмір елементів для оптимізації перегляду на мобільних пристроях, планшетах та великих моніторах.

CSS також сприяє підвищенню доступності та семантики веб-сторінок. Призначаючи класи та ідентифікатори елементам HTML, розробники можуть застосовувати певні стилі до різних елементів, покращуючи розуміння структури сторінки та підвищуючи доступність для користувачів з обмеженими можливостями.

Таким чином, CSS дозволяє розробникам створювати візуально привабливі, функціональні та зручні для користувача веб-сайти і додатки. Її використання суттєво впливає на зовнішній вигляд, користувацький досвід і взаємодію з

відвідувачами. Впровадження CSS є важливим кроком на шляху до професійної та сучасної веб-розробки.

SCSS (Sassy CSS) і Sass (Syntactically Awesome StyleSheets) - це мови сценаріїв препроцесора, які використовуються для створення таблиць стилів CSS. Вони надають додаткові можливості та покращення у порівнянні з традиційним CSS, роблячи процес написання та підтримки таблиць стилів більш ефективним та гнучким [2].

І SCSS, і Sass використовують підмножину синтаксису CSS, а це означає, що коректний код CSS також є коректним кодом SCSS/Sass. Однак, вони вводять нові можливості, такі як змінні, вкладеність, міксини, успадкування та функції, що дозволяє створювати більш модульні та багаторазові таблиці стилів.

Основні відмінності між SCSS та Sass пов'язані з синтаксисом. SCSS використовує синтаксис, схожий на CSS, з фігурними дужками та крапкою з комою, що полегшує перехід на SCSS для розробників, які вже знайомі з CSS. З іншого боку, Sass використовує більш лаконічний синтаксис, заснований на відступах, без фігурних дужок і крапки з комою, який деякі розробники вважають більш читабельним і елегантним.

<pre> //SASS記法 SASS div margin:-0.0.10px display:-block img width:-50px height:-50px; //SCSS記法 SCSS <div>{ margin:-0.0.10px; display:-block; } img{ width:-50px; height:-50px; } </div></pre>	<pre> CSS 335 336 ▾ div·{ 337 ··margin:-0.0.10px; 338 ··display:-block; 339 } 340 ▾ div·img·{ 341 ··width:-50px; 342 ··height:-50px; 343 } </pre>
--	--

Рисунок 2.2 – Відмінності препроцесорів

І в SCSS, і в Sass код обробляється препроцесором, який компілює код

SCSS/Sass у стандартний CSS, зрозумілий веб-браузерам. Це дозволяє розробникам писати складні таблиці стилів з розширеними можливостями, а потім генерувати оптимізовані CSS-файли для виробничого використання [7].

Загалом, SCSS та Sass пропонують потужні інструменти для організації та підтримки таблиць стилів CSS, зменшення надмірності та покращення повторного використання коду, що може підвищити ефективність та ремонтпридатність проектів з веб-розробки.

2.4 Використання інструментів для автоматизації Gulp

Gulp – популярний інструмент для автоматизації повторюваних завдань у процесі розробки веб-сайтів. Він впорядковує та автоматизує різні завдання розробки, що робить його одним з найпоширеніших інструментів у галузі.

За допомогою Gulp розробники можуть створювати і виконувати широкий спектр завдань, включаючи збір і мінімізацію файлів CSS і JavaScript, оптимізацію зображень, автоматичне оновлення веб-сторінок при зміні файлів, запуск тестів і багато іншого [10].

Основна перевага використання Gulp полягає в тому, що він дозволяє розробникам зосередитися на написанні коду замість того, щоб витратити надмірний час на рутинні завдання. Gulp автоматично виконує необхідні операції з файлами, дозволяючи розробникам ефективно працювати над своїми проектами.

Крім того, Gulp є дуже розширюваним, надаючи можливість включати різні плагіни для додаткової функціональності. Ці плагіни можуть включати такі функції, як автоматизований перегляд коду, генерація документації, розгортання на реальних серверах тощо.

Таким чином, Gulp – це надійний інструмент, який дозволяє розробникам ефективно керувати та автоматизувати процес розробки веб-сайтів. Автоматизуючи рутинні завдання, Gulp підвищує продуктивність, дозволяючи розробникам виконувати свою роботу швидко та ефективно [11].

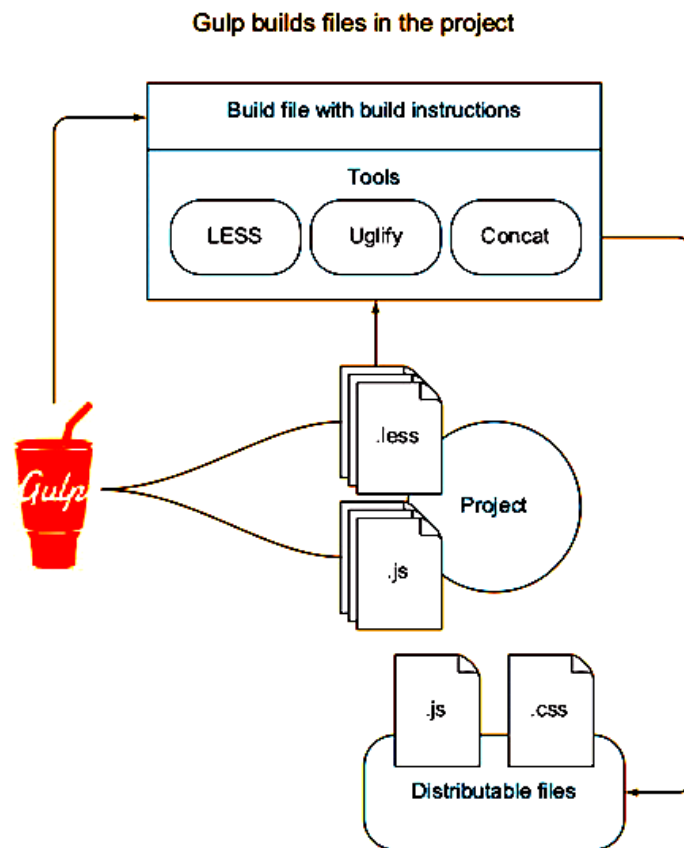


Рисунок 2.3 – Інструмент збірки Gulp

```
// Підключення необхідних модулів
const gulp = require('gulp');
const sass = require('gulp-sass');
const minifyCSS = require('gulp-clean-css');
const concat = require('gulp-concat');
// Завдання для компіляції SCSS в CSS
gulp.task('compile-scss', function () {
  return gulp.src('src/scss/**/*.scss') // Вихідні файли SCSS
    .pipe(sass()) // Компіляція SCSS в CSS
    .pipe(gulp.dest('dist/css')); // Збереження скомпільованих CSS-
    файлів
});
```

У цьому коді ми спочатку підключаємо необхідні модулі Галпа, такі як `gulp`, `gulp-sass`, `gulp-clean-css` та `gulp-concat`. Потім ми визначаємо два завдання: `compile-scss` та `minify-css`.

Завдання `compile-scss` відповідає за компіляцію SCSS-файлів в CSS-файли. Ми вказуємо вхідні файли SCSS, застосовуємо компіляцію за допомогою `gulp-sass` та зберігаємо скомпільовані CSS-файли у відповідній директорії.

Завдання `minify-css` зливає всі CSS-файли в один файл за допомогою `gulp-concat`, мініфікує цей файл за допомогою `gulp-clean-css` і зберігає мініфікований CSS-файл.

Нарешті, за допомогою `gulp.task('default', gulp.series('compile-scss', 'minify-css'))` ми встановлюємо завдання за замовчуванням, яке буде виконувати обидва завдання послідовно при запуску `gulp`.

Цей код є простим прикладом використання Галпа, і ви можете розширити його функціональність, додавши інші плагіни та завдання відповідно до вашого проекту.

2.5 Пакетний менеджер npm

На додаток до вбудованих і кастомних модулів, що надаються Node.js, існує величезна кількість бібліотек, фреймворків і утиліт, розроблених сторонніми розробниками, які можна використовувати в проектах. Прикладами таких інструментів є `express`, `grunt`, `gulp` та інші. Ці ресурси доступні в середовищі Node.js. Щоб полегшити використання цих сторонніх рішень, вони поширюються у вигляді пакетів. Пакет, по суті, охоплює набір функціональних можливостей. Для автоматизації процесу встановлення та оновлення пакунків використовуються системи керування пакунками або менеджери. Node Package Manager (NPM) є одним з таких інструментів, який використовується безпосередньо в Node.js для цієї мети. NPM постачається з попередньо встановленим Node.js, що усуває необхідність у будь-яких додаткових установках. Однак, при бажанні можна оновити NPM до останньої версії [21].

Node Package Manager (NPM) слугує зручним інструментом, який дозволяє розробникам JavaScript ефективно керувати залежностями (пакетами) у своїх проектах.

У сфері програмування розробники часто використовують готові рішення, створені іншими розробниками, такі як бібліотеки, фреймворки та інші корисні

інструменти. Ці рішення, як правило, упаковані для зручності використання. Процес окремого завантаження та встановлення кожного пакунка може бути складним і трудомістким. Саме тут на допомогу приходить npm.

Npm дозволяє розробникам швидко і без зайвих зусиль встановлювати пакунки з великого сховища пакунків. Вказавши необхідні пакунки для проекту, npm автоматично виконує завантаження та встановлення всіх необхідних залежностей.

Крім того, npm надає розробникам можливість керувати версіями пакунків, виконувати оновлення та вирішувати конфлікти, які можуть виникнути через використання різних версій залежностей. Він пропонує зручне робоче середовище, де пакунками можна легко керувати, забезпечуючи ефективну розробку проектів.

У цій роботі було використано декілька плагінів npm для поліпшення розробки та оптимізації веб-сайту. Ось короткий опис кожного з них:

1 SCSS: Плагін SCSS дозволяє використовувати препроцесор Sass (Syntactically Awesome Style Sheets) для покращення написання CSS. Sass надає розширені можливості, такі як змінні, міксіни, вкладеність, що полегшують стилізацію веб-сайту [7].

2 MinifyJS: Цей плагін дозволяє мініфікувати (стискувати) JavaScript-файли, зменшуючи їх розмір. Мініфікація забезпечує видалення зайвих пробілів, коментарів та інших незначних деталей, що дозволяє прискорити завантаження сторінки та зменшити використання мережевого трафіку.

3 Layout: Цей плагін надає зручність у розміщенні та організації елементів на веб-сторінці. Він надає функціонал для швидкого створення та керування розташуванням блоків, сітки, адаптивного дизайну та інших аспектів сторінки.

4 CriticalCSS: Цей плагін допомагає оптимізувати завантаження сторінки шляхом генерації критичного CSS. Критичний CSS є мінімальною необхідною частиною CSS, необхідною для відображення видимого контенту на

початковому екрані. Він дозволяє швидше відображення сторінки та покращує загальний час завантаження.

5 Використання цих плагінів прм сприяло покращенню розробки, стилізації, оптимізації та ефективності веб-сайту, що був розроблений в рамках даної роботи.

2.6 Мова програмування JavaScript

JavaScript - інтерпретована мова програмування, яка є однією з реалізацій мови ECMAScript і базується на концепції прототипів, яка походить з мови Self. Зараз JavaScript широко використовується для створення сценаріїв, що керують поведінкою веб-сторінок у браузерях, а також як мова сценаріїв для доступу до об'єктів програм. Деякі програми включають в себе вбудований інтерпретатор JavaScript, що дозволяє розширювати їх функціональність без зміни самої програми. Мову програмування JavaScript розробила компанія Netscape спільно з Sun Microsystems для створення інтерактивних HTML-документів. Основні сфери застосування JavaScript включають:

- 1 Створення динамічних веб-сторінок, які можуть змінюватися після завантаження документа.
- 2 Перевірка правильності заповнення форм користувачами перед їх відправкою на сервер.
- 3 Вирішення завдань на стороні клієнта за допомогою сценаріїв, а також інші сфери, де можна створювати додатки, що працюють як на клієнтському, так і на серверному боці.

При розробці таких додатків використовується спеціальне ядро, що містить визначення стандартних об'єктів. Клієнтські програми виконуються у браузері на комп'ютері користувача. Програма або сценарій на мові JavaScript обробляється вбудованим інтерпретатором, що знаходиться в браузері. Важливо забезпечити коректне виконання написаних сценаріїв у будь-якому браузері. Код JavaScript

включається в HTML-код сторінки і виконується інтерпретатором, що знаходиться вбудованим у браузер [11].

JavaScript на стороні клієнта. JavaScript на стороні клієнта використовується для взаємодії з користувачем і керування поведінкою веб-сторінок у браузері. Його основні завдання охоплюють.

Маніпуляції з DOM:

1 JavaScript може змінювати структуру та вміст веб-сторінки, додаючи, видаляючи або модифікуючи елементи, змінюючи стилі, обробляючи події (наприклад, кліки миші або введення користувача) та оновлюючи вміст без необхідності повного перезавантаження сторінки.

2 Перевірка форм: JavaScript може перевіряти дані, введені користувачем у формах, перед передачею їх на сервер. Це забезпечує належну обробку даних і дозволяє сповіщати про помилки без необхідності надсилати дані на сервер.

3 Взаємодія з сервером: JavaScript може робити асинхронні запити до сервера, наприклад, використовуючи технологію AJAX, щоб надсилати або отримувати дані без перезавантаження сторінки. Це дозволяє створювати динамічні сторінки, які оновлюють або відображають нову інформацію без повного оновлення.

4 Робота з файлами cookie та локальним сховищем: JavaScript може читати, записувати і видаляти файли cookie, що зберігаються на стороні клієнта, які використовуються для зберігання даних між взаємодіями з веб-сторінкою. Крім того, JavaScript може використовувати локальне сховище для безпосереднього зберігання даних на стороні клієнта.

5 Анімація та візуалізація: JavaScript дозволяє створювати анімацію, переходи між сторінками, динамічні ефекти та інтерактивну візуалізацію даних на веб-сторінках.

6 JavaScript на стороні клієнта є надійним інструментом, що дозволяє розробляти багатофункціональні та інтерактивні веб-додатки, які працюють безпосередньо в браузері користувача [16].

JavaScript переваги. Ось основні переваги мови програмування:

1 JavaScript є необхідним для сучасних браузерів і підтримується всіма ними.

2 Використання JavaScript не потребує спеціальних навичок - будь-який фахівець може використовувати плагіни та скрипти, написані на JavaScript.

3 Мова має багато корисних функціональних можливостей.

4 JavaScript постійно розвивається і оновлюється.

5 Мова має перспективи використання в процесі навчання програмування та інформатики.

6 JavaScript дозволяє створювати привабливі інтерфейси, включаючи компоненти перетягування і повзунки, для багатфункціональних сайтів.

7 Використання JavaScript є необхідним для розробки веб-сайтів із високою функціональністю.

8 JavaScript підтримується всіма популярними браузерами і інтегрується з HTML, CSS і серверною частиною (backend).

9 Використання JavaScript забезпечує швидкість роботи і продуктивність веб-додатків.

JavaScript дозволяє частково обробляти веб-сторінки без звернення до сервера на комп'ютерах користувачів. Це зберігає час і трафік, зменшує навантаження на сервер.

Існує потужна інфраструктура (екосистема) для JavaScript. У перші 10 років такої інфраструктури не було навіть уявлення. Згодом кількість готових рішень, доступних в відкритому доступі, значно зросла, що зробило роботу з JavaScript та його фреймворками задоволенням.

JavaScript має простоту і раціональність в застосуванні. Прості завдання можна вирішити за 5 хвилин, не витрачаючи зайвий час. Для складних завдань існують різні варіанти розв'язання, які можна підібрати і адаптувати до потреб.

JavaScript забезпечує зручність для користувачів інтерфейсів. Це стосується заповнення форм, вибору дій, активації кнопок, перевірки введення, реагування на наведення/кліки миші та інших дій. Це значно підвищує рівень зручності використання (юзабіліті) для користувачів [5].

JavaScript обмеження:

1 JavaScript не можна вважати повноцінною мовою програмування через відсутність кількох важливих функцій:

2 Знижена безпека: Доступність вихідного коду популярних скриптів призводить до зниження рівня безпеки.

3 Численні помилки: JavaScript часто стикається з невеликими, але надокучливими помилками на різних етапах роботи. Хоча більшість цих помилок можна виправити, їх наявність свідчить про нижчий рівень професіоналізму порівняно з іншими мовами програмування.

4 Невідповідності в підтримці браузерів: Різні браузери інтерпретують JavaScript по-різному. Ця розбіжність в інтерпретації, спричинена різними механізмами верстки, може призвести до відмінностей у функціональності та інтерфейсі. Крім того, доступ до об'єктів може відрізнитися в різних браузерах.

5 Доступність файлів: Файли JavaScript завантажуються на клієнтську машину, що дозволяє будь-кому читати і повторно використовувати код.

6 Вимкнення JavaScript: Якщо JavaScript вимкнено у браузері, весь JavaScript-код не виконується.

7 Непридатність для мережевих додатків: JavaScript не має підтримки мережевих додатків, що обмежує його застосування в цій сфері.

8 Важливо зазначити, що ці обмеження стосуються конкретних аспектів JavaScript і не зменшують його цінності як мови програмування для веб-розробки та створення динамічних інтерфейсів [4].

2.7 PHP-шаблонізатор Twig

Twig є шаблонізатором, створеним для мови програмування PHP, який дозволяє розробникам зручно відокремлювати логіку та представлення веб-сторінок, спрощуючи розробку та підтримку веб-додатків [23].

Основна мета Twig полягає в наданні можливості шаблонування даних, що

означає відокремлення логіки додатку від його представлення. Замість прямого включення PHP-коду в HTML-файли, Twig дозволяє використовувати спеціальні теги та синтаксис для вставки змінних, умовних виразів, циклів та інших конструкцій.

Основні переваги використання Twig включають:

1 *Чистий і зрозумілий синтаксис:* Twig пропонує простий і зрозумілий синтаксис, що полегшує читання та редагування шаблонів.

2 *Розділення логіки та представлення:* Twig дозволяє відокремити логіку програми від відображення даних, що полегшує підтримку та повторне використання коду.

3 *Безпека:* Twig включає вбудовану систему безпеки, яка захищає веб-додатки від атак типу Cross-Site Scripting (XSS).

4 *Велика функціональність:* Twig надає багато корисних функцій, таких як фільтри для форматування даних та розрахунку хеш-значень.

5 *Розширюваність:* Twig може бути розширений за допомогою власних розширень та функцій, що дозволяє розробникам розширити можливості шаблонізатора за своїми потребами.

Twig є одним з найпопулярніших шаблонізаторів для PHP, який широко використовується у веб-розробці. Він допомагає зробити код більш читабельним, підтримуваним та безпечним, а також сприяє ефективній роботі над проектами.

Щоб створити сайт за допомогою Twig, вам знадобиться наступне:

1 *Встановлення:* По-перше, вам потрібно встановити Twig, що можна легко зробити за допомогою Composer, менеджера пакетів для PHP. Встановлення передбачає запуск команди в терміналі або командному рядку.

2 *Структура шаблону:* Після встановлення ви можете створити папку для зберігання ваших шаблонів. Рекомендується мати спеціальну папку, наприклад, "templates", де зберігатимуться всі шаблони.

3 *Синтаксис Twig:* Twig має власний синтаксис для створення шаблонів. Він включає спеціальні теги, фільтри, змінні та функції, які дозволяють вставляти та обробляти дані [23].

4 *Робота з даними:* Twig дозволяє передавати дані в шаблон з PHP-коду. Ви можете використовувати змінні, об'єкти, масиви та інші структури даних для відображення вмісту на веб-сторінці.

5 *Керування потоком:* Twig забезпечує умовне відображення, цикли та інші конструкції керування потоком. Це дозволяє вам змінювати поведінку шаблону на основі умов та інших факторів.

6 *Використання фільтрів і функцій:* Twig постачається з вбудованими фільтрами та функціями для форматування даних, маніпуляцій з рядками, обчислення значень тощо.

7 *Спадкування та включення шаблонів:* Twig підтримує успадкування шаблонів, що дозволяє вам створити базовий шаблон і розширити його для відображення певних сторінок. Крім того, ви можете включати інші шаблони в свої шаблони для повторного використання коду.

8 *Кешування:* Twig підтримує механізм кешування, який дозволяє зберігати скомпільовані шаблони, підвищуючи продуктивність сайту.

Twig – це потужний і гнучкий інструмент для розробки веб-сайтів на PHP. Він ефективно відокремлює логіку від представлення і надає широкі функціональні можливості для роботи з шаблонами. За допомогою Twig ви можете створювати чистий, читабельний і легко підтримуваний код для вашого веб-додатку [17].

```
<nav>
  <ul>
    {% for item in menuItems %}
      <li>
        <a href="{{ item.url }}">{{ item.label }}</a>
      </li>
    {% endfor %}
  </ul>
</nav>
```

У цьому прикладі ми використовуємо цикл `for` для перебору елементів `menuItems`, який є масивом об'єктів з властивостями `url` (URL-адреса сторінки) та `label` (підпис елемента меню). Для кожного елемента ми генеруємо `` елемент списку з посиланням `<a>`, де URL та підпис отримуються з об'єкта `item`.

РОЗДІЛ 3. ВИКОРИСТАННЯ МЕТОДІВ СПРОЩЕННЯ РОЗРОБКИ ВЕБ-САЙТУ

3.1 Середовище розробки Visual Studio Code

Visual Studio Code (VS Code) – це безкоштовний, легкий і потужний редактор коду, створений компанією Microsoft для веб-розробки та програмування. Він дуже популярний серед розробників завдяки своїм широким можливостям та функціям.

Ось деякі з ключових можливостей, які пропонує VS Code:

1 Розмітка та підсвічування синтаксису: VS Code підтримує різні мови програмування та мови розмітки, забезпечуючи підсвічування синтаксису для покращення читабельності та редагування коду.

2 Розширення та плагіни: VS Code пропонує широкий спектр розширень, які дозволяють налаштовувати та розширювати функціональність редактора. Розробники можуть встановлювати додаткові інструменти для підтримки певних мов програмування, фреймворків, автоматизації завдань тощо.

3 Інтегрована система контролю версій: VS Code легко інтегрується з системами контролю версій, такими як Git, що дозволяє розробникам легко керувати своїми проектами та відстежувати зміни.

4 Вбудований термінал: VS Code має вбудований термінал, що дозволяє розробникам виконувати команди безпосередньо в середовищі розробки. Це корисно для таких завдань, як запуск сервера, виконання скриптів та інших командних операцій [24].

5 Налагодження: VS Code надає вбудовані можливості налагодження, що дозволяють розробникам налагоджувати код безпосередньо в редакторі. Це полегшує виявлення помилок, аналіз змінних і моніторинг виконання програми.

6 Співпраця та спільне використання коду: VS Code підтримує спільну розробку, дозволяючи декільком розробникам одночасно працювати над кодом, обмінюватися коментарями та вносити спільні зміни.

Ці функції, разом з багатьма іншими, сприяють популярності Visual Studio Code як інструменту розробки програмного забезпечення. Він доступний на багатьох платформах і користується перевагами активної спільноти розробників, яка постійно вдосконалює та розширює його можливості.

У Visual Studio Code надається широка підтримка майже всіх основних мов програмування. Хоча деякі мови, такі як JavaScript, TypeScript, CSS та HTML, входять до складу пакету, додаткові мовні розширення можна знайти на ринку VS Code Marketplace.

3.2 Розробка структури веб-сайту

На цьому етапі визначається логічна структура веб-сайту. Створюється схема навігації, визначаються основні розділи та підрозділи, їх взаємозв'язок та ієрархія. Важливо ретельно розробити структуру веб-сайту, оскільки це визначає організацію інформації та навігаційну логіку, що впливає на користувацький досвід та ефективність сайту.

Структура веб-сайту "Urban" включатиме наступні розділи:

1. Про нас:
 - Хто ми такі: У цьому розділі користувачі зможуть дізнатися про наш заклад, нашу історію та цінності.
 - Що ми пропонуємо: Тут ми розповімо про нашу концепцію гастрономії та особливості наших страв.
 - Категорії страв: В цій частині сайту будуть представлені головні категорії наших страв, наприклад, бургери, піца, барне меню, закуски. Користувачі зможуть дізнатися про наш розмаїтий асортимент та знайти те, що їм до вподоби.
2. Меню:
 - Основне меню: У цьому розділі буде представлено наше основне меню з усіма нашими стравами та їх описами.

- Сніданки: Тут користувачі зможуть ознайомитися з нашими смачними сніданками, включаючи різноманітні варіанти та склад інгредієнтів.

- Обіди: У цій категорії будуть представлені наші смачні обіди з різними стравами, які задовольнять різні смакові уподобання.

- Барне меню: Для любителів апетитних закусок та освіжаючих напоїв ми пропонуємо барне меню з різноманітними варіантами.

3. Доставка:

- Сервіси доставки: Користувачі матимуть можливість вибрати зручний сервіс доставки і замовити наші страви з доставкою прямо до свого дому або офісу.

4. Галерея:

- Фото закладу: У цьому розділі ми покажемо фотографії нашого закладу, його атмосфери та дизайну, щоб користувачі могли отримати візуальне уявлення про наш ресторан.

- Презентація нових страв: Ми будемо регулярно оновлювати галерею з фотографіями наших нових страв та спеціальних пропозицій, щоб залучити користувачів та заохотити їх спробувати щось нове.

- Різні заходи: Тут ми будемо ділитися фотографіями з різних заходів та подій, які відбуваються у нашому закладі, створюючи атмосферу спільноти та залучаючи наших клієнтів.

5. Контакти:

- Інформація про заклад: В цьому розділі користувачі знайдуть інформацію про розташування нашого закладу, години його роботи, контактні дані та соціальні мережі, де можна знайти додаткову інформацію про нас.

- Місцезрештування на карті: Щоб допомогти користувачам знайти нас, ми надаємо місцезрештування нашого закладу на карті з можливістю отримати маршрут до нього.

6. Футер веб-сайту "Urban" буде містити наступну інформацію:

- Навігація по сайту: В футері будуть розташовані посилання на основні розділи сайту, такі як "Про нас", "Меню", "Доставка", "Галерея", "Контакти". Це

дозволить користувачам швидко перейти до потрібного розділу сайту з будь-якої сторінки.

- Контактні дані: У футері будуть вказані контактні дані закладу, такі як телефон, електронна пошта або адреса. Це надасть користувачам можливість зв'язатися з закладом безпосередньо з будь-якої сторінки сайту.
- Логотип: У футері буде розміщений логотип закладу, який служить візуальним ідентифікатором сайту. Це допоможе користувачам легко розпізнати бренд і пов'язати його зі сторінками сайту.

Футер веб-сайту "Urban" виконує важливу роль у навігації користувачів, забезпечує доступ до важливих розділів сайту та контактної інформації, а також покращує візуальне сприйняття сайту завдяки логотипу.

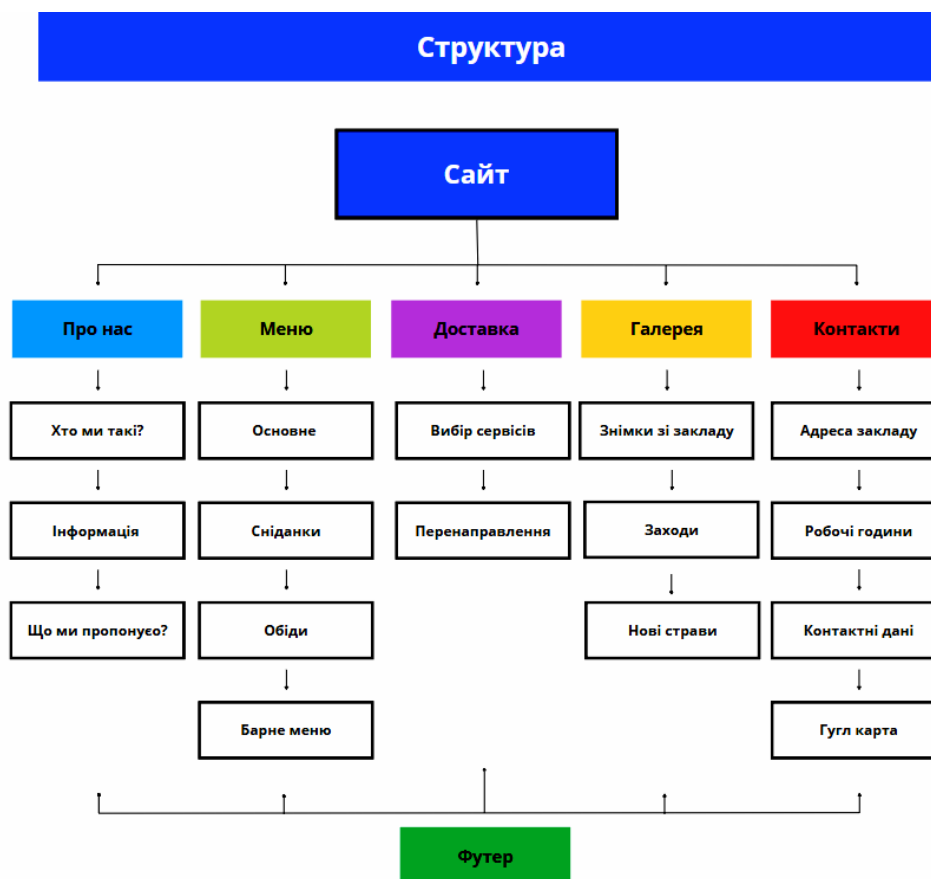


Рисунок 3.1 – Структура сайту

Ця структура сайту допоможе користувачам легко орієнтуватись у вмісті та знайти необхідну інформацію, меню та послуги, що ми пропонуємо.

3.2 Етапи створення макету веб-сайту

На цьому етапі розробки веб-сайту "Urban" створюється візуальний макет, який допоможе визначити зовнішній вигляд та організацію інтерфейсу сайту. Для створення макету використовується векторний онлайн-сервіс розробки інтерфейсів та прототипування Figma.

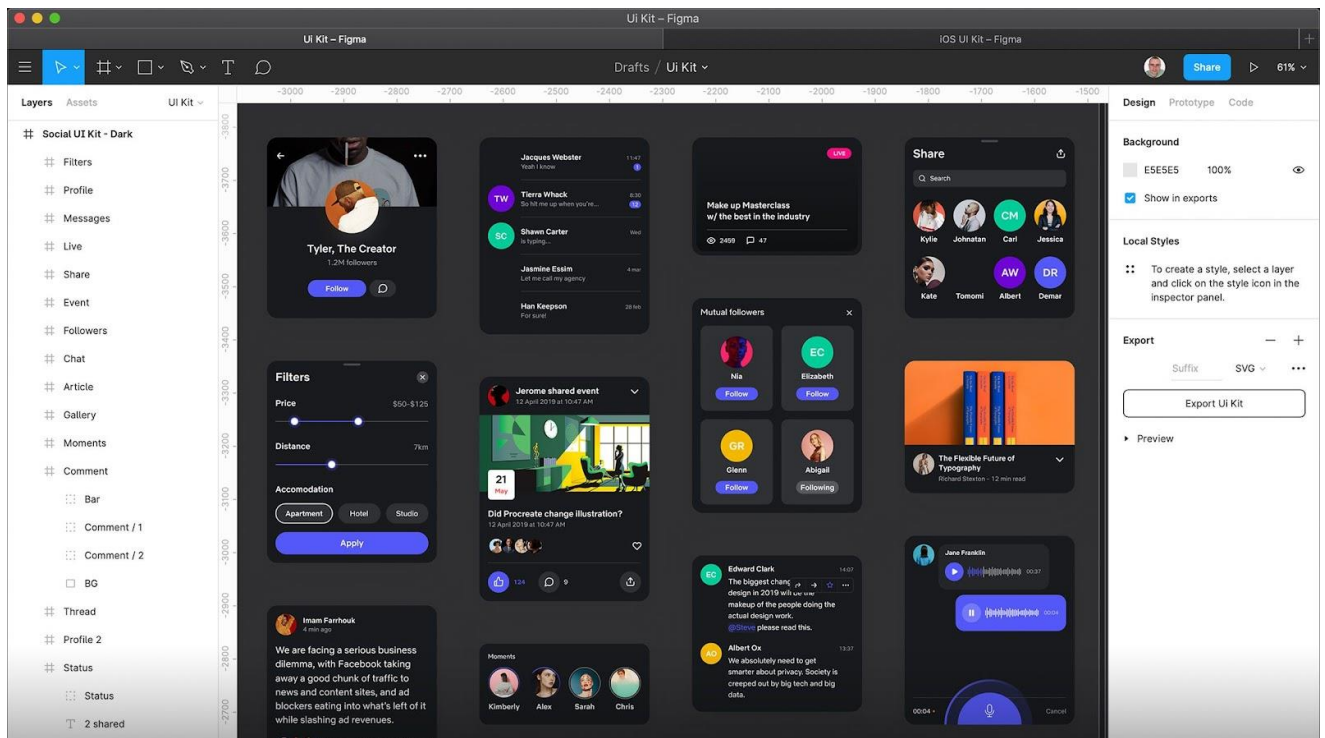


Рисунок 3.2 – Головна сторінка застосунку Figma

Головною метою створення макету є візуалізація концепції та структури веб-сайту. На початку процесу розробки макету, дизайнер вивчає вимоги та потреби користувачів, аналізує конкурентів і розробляє концепцію дизайну, яка відповідає цілям та цінностям "Urban".

Дизайн головної сторінки має привабливий та привертаючий вигляд, який відображає основну інформацію про заклад і стимулює користувачів до подальшого дослідження сайту. На головній сторінці розміщуються ключові елементи, такі як логотип "Urban", навігаційне меню, відео з привабливими зображеннями, а також короткий опис закладу та його основних послуг.

Для внутрішніх сторінок сайту розробляються макети, які відображають

розташування різних елементів, наприклад, заголовків, текстового контенту, зображень, кнопок та інших важливих компонентів. Дизайнер також встановлює колірну палітру, шрифти та інші елементи, що сприяють створенню єдиної стильової концепції сайту.

Створений макет перевіряється командою розробників та замовником проекту для отримання зворотного зв'язку та внесення необхідних змін. Після узгодження макету він стає основою для подальшої розробки функціональності та імплементації дизайну веб-сайту.

Створення детального та естетичного макету є важливим етапом у процесі розробки веб-сайту "Urban", оскільки він допомагає зрозуміти, як будуть виглядати різні сторінки та як користувачі будуть взаємодіяти з сайтом.

3.3 Реалізація функціональності веб-сайту

На цьому етапі реалізуємо необхідну функціональність веб-сайту "Urban". Основною метою цього етапу є створення інтерактивних елементів та реалізація різноманітних функцій, що дозволять користувачам ефективно взаємодіяти з сайтом.

Один з важливих аспектів розробки функціональності – це створення інтерактивних елементів. Це можуть бути кнопки, посилання, меню, слайдери, форми зворотного зв'язку та інші елементи, які забезпечують активну взаємодію користувача з сайтом. Розробимо логіку роботи цих елементів, запрограмуємо їх взаємодію з користувачем.

Розробка функціональності веб-сайту "Urban" є складним процесом, який вимагає великої уваги до деталей, знання програмування та розуміння вимог користувачів. Програмісти працюють у команді, спілкуються з дизайнерами та іншими спеціалістами для досягнення бажаного результату - функціонального, зручного та ефективного веб-сайту "Urban".

У рамках функціональності веб-сайту "Urban" передбачаються наступні

можливості:

1. Розділи сайту: Користувачі зможуть переключатись між різними розділами меню, такими як "Про нас", "Меню", "Галерея". Ці розділи будуть клікабельними, що дозволить користувачам швидко переходити до бажаного розділу.

2. Відео: На сайті буде розміщено відео, яке демонструє суть закладу. Користувачі зможуть переглядати це відео, отримуючи уявлення про атмосферу та особливості закладу.

3. Категорії страв: Категорії страв також будуть клікабельними, що дозволить користувачам переходити до конкретної категорії та переглядати страви, які вона містить.

4. Фотогалерея: Сайт буде мати фотогалерею, в якій користувачі зможуть переглядати зображення закладу. Крім того, надана можливість збільшити фото для більш детального огляду.

5. Гугл-карта: На сайті буде вбудована гугл-карта, на якій буде показане місцезнаходження закладу. Користувачі зможуть переходити до цієї карти та використовувати її для прокладання маршруту до закладу.

6. Легке пересування по розділам страв: Сайт буде мати зручну навігацію, що дозволить користувачам легко пересуватись між різними розділами страв та швидко знайти потрібну інформацію.

7. Збільшення фото страв: Користувачі зможуть збільшувати фотографії страв для детального ознайомлення з закладом, його послугами та замовлення необхідних страв.

3.4 Верстка та оформлення

Цей розділ є важливим кроком у розробці веб-сайту в рамках моєї кваліфікаційної роботи. На цьому етапі я використовую мову розмітки HTML та каскадні таблиці стилів CSS, щоб створити виглядову форму мого веб-сайту.

За допомогою мови розмітки HTML структуруємо та розміщуємо елементи на моїх веб-сторінках. Використовуємо теги, такі як заголовки, абзаци, списки, таблиці та посилання, щоб організувати мої сторінки. HTML дозволяє мені створювати ієрархічну структуру мого веб-сайту та визначати зміст кожної сторінки.

CSS допомагає в оформленні мого веб-сайту. За допомогою каскадних таблиць стилів встановлюємо кольори, шрифти, розташування, розміри та інші стилі для елементів на моїх сторінках. Використовуємо CSS, щоб надати сторінкам відповідний дизайн та оформлення.

На цьому етапі нам потрібно HTML та CSS для створення розмітки сторінок, розташування елементів, задання стилів та оформлення контенту. Потрібно врахувати вимоги дизайну, кольорову палітру, шрифти та інші деталі, щоб створити привабливий та функціональний дизайн мого веб-сайту.

Всі подальші дії щодо розробки веб-сайту "Urban" будуть виконані в редакторі коду Visual Studio Code. Цей редактор надає зручне та потужне середовище для редагування та відлагодження мого коду. Використання Visual Studio Code допомагає зберігати та організовувати проект, спрощує процес розробки та поліпшує продуктивність.

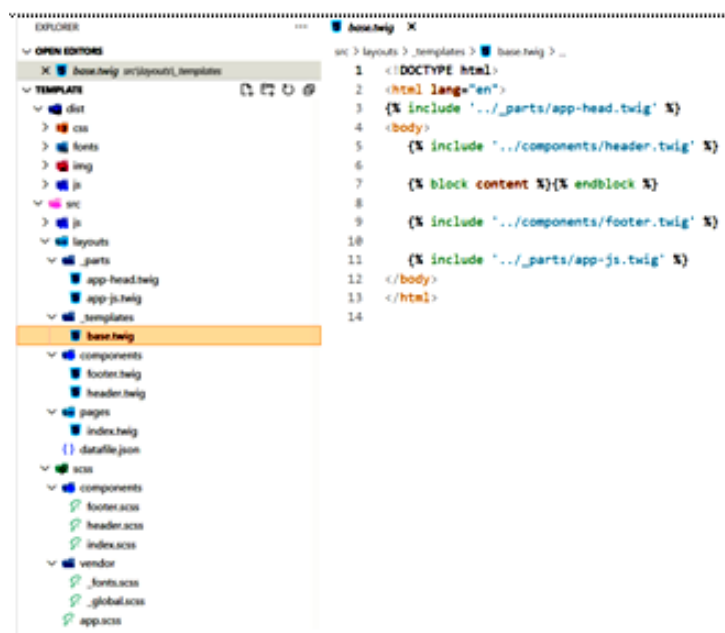


Рисунок 3.3 – Структура проекту в VS Code

Для збірки фронтенду веб-сайту, використовуємо інструмент Gulp. Gulp дозволяє автоматизувати рутинні задачі розробки, такі як компіляція CSS та JavaScript файлів, оптимізація зображень, побудова та оптимізація файлів для виробничого середовища. Використання Gulp допомагає мені зекономити час та зусилля, забезпечує ефективну роботу над моїм проектом.

Щодо шаблонізатора, використаємо Twig для розробки веб-сайту "Urban". Twig є потужним та гнучким шаблонізатором, який дозволяє використовувати шаблони для створення динамічного контенту на моїх сторінках. Використання Twig спрощує процес створення та управління різноманітними компонентами мого веб-сайту, такими як заголовки, навігаційні меню, списки страв та інші. Шаблонізатор Twig допомагає створити ефективну та повторно використововану кодову базу для мого проекту.

За допомогою шаблонізатора Twig можна використовувати компонентний підхід у розробці веб-сайту. Це означає, що я буду розбивати різні блоки сторінки на окремі шаблони, які потім можна буде підключати та використовувати на різних сторінках.

Компонентний підхід дозволяє розділити веб-сторінку на незалежні компоненти або блоки, які мають свою функціональність та стиль. Кожен компонент може мати свій власний шаблон зі своєю логікою та стилями. Це дозволяє зручно керувати структурою та макетом веб-сайту, а також полегшує повторне використання коду.

Завдяки компонентному підходу з використанням Twig, я зможу створювати шаблони для загальних блоків, таких як хедер, футер, меню, галерея, тощо, і використовувати їх у багатьох різних сторінках. Це забезпечить однорідність дизайну та функціональності на всьому веб-сайті і значно спростить процес розробки та підтримки.

Компонентний підхід забезпечує більшу модульність, гнучкість та перевикористання коду, що в свою чергу сприяє збереженню часу та зусиль при розробці веб-сайту. Крім того, це підвищує читабельність коду і полегшує співпрацю з іншими розробниками.

Верстка нашого веб-сайту є семантичною, що означає, що всі блоки розмічені згідно їх семантики та призначення. Семантична верстка полягає в використанні HTML-тегів згідно їх семантичного значення для кращого розуміння структури та змісту веб-сторінки як для розробників, так і для пошукових систем [18].

Наприклад, використання тегів `<header>`, `<nav>`, `<main>`, `<footer>` відображає структурні елементи сторінки. Теги `<article>`, `<section>`, `<aside>` використовуються для розмітки основного змісту сторінки. Теги `<h1>`-`<h6>` використовуються для заголовків залежно від їх важливості та ієрархії.

Семантична верстка допомагає покращити доступність веб-сайту, зробити його зрозумілішим для читачів, а також підвищити рейтинг сторінки в пошукових системах, оскільки вони краще розуміють структуру та зміст сторінки за допомогою семантичних тегів.

Таким чином, використання семантичної верстки важливо для покращення якості веб-сайту і забезпечення його оптимальної інтерпретації технологічними засобами та користувачами.

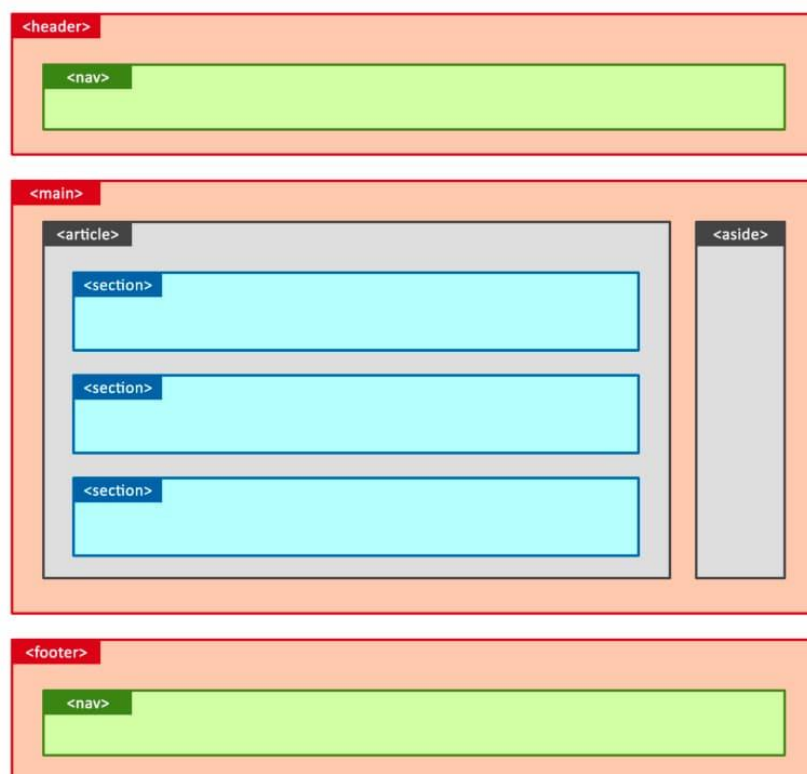


Рисунок 3.4 – Семантична HTML-розмітка

Принцип верстки "mobile first" означає, що спочатку розробляється мобільна версія веб-сайту, а потім вже адаптується для більших екранів, таких як планшети та настільні комп'ютери. Цей підхід став дуже актуальним у сучасному веб-розробці.

Початкова розробка мобільної версії дозволяє зосередитись на основних функціях та вмісті сайту, які повинні бути доступні на будь-якому пристрої. Такий підхід враховує зростання використання мобільних пристроїв та вимоги користувачів до зручності та швидкості завантаження веб-сторінок на мобільних пристроях.

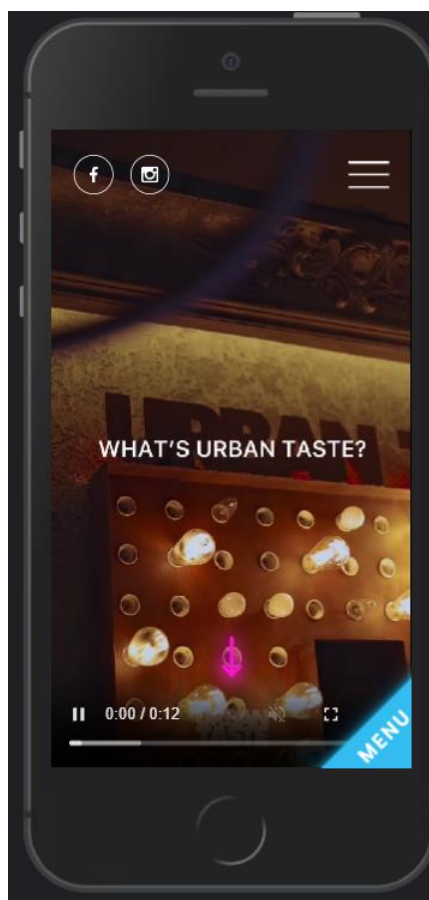


Рисунок 3.5 – Головна сторінка сайту

У першому головному розділі мого веб-сайту "Urban" розташовані такі елементи. В хедері сайту розміщені посилання на соціальні мережі, що дозволяє користувачам швидко переходити на сторінки закладу в соціальних мережах для отримання більш детальної інформації. Також в хедері розташоване меню, в якому доступні розділи сайту, що дозволяє користувачам легко перемикатися по

різних секціях веб-сайту.

Під хедером розміщений головний банер у вигляді відео, яке не лише розповідає про суть закладу, але й ілюструє його атмосферу та особливості за допомогою різних картинок. Цей банер привертає увагу відвідувачів і створює перший враження про веб-сайт "Urban".

HTML код для хедера:

```
<header class="header">
  <nav class="nav">
    <ul class="nav-list">
      <li class="nav-item"><a href="#" class="nav-link">Про нас</a></li>
      <li class="nav-item"><a href="#" class="nav-link">Меню</a></li>
      <li class="nav-item"><a href="#" class="nav-link">Доставка</a></li>
      <li class="nav-item"><a href="#" class="nav-link">Галерея</a></li>
      <li class="nav-item"><a href="#" class="nav-link">Контакти</a></li>
    </ul>
  </nav>
  <div class="social-icons">
    <a href="#" class="social-icon"><i class="fab fa-facebook"></i></a>
    <a href="#" class="social-icon"><i class="fab fa-instagram"></i></a>
  </div>
</header>
```

CSS код для стилізації хедера:

```
.header {
  background-color: #000;
  padding: 10px;
  color: #fff;
}
.nav {
  display: flex;
  justify-content: space-between;
  align-items: center;
```

```

}
.nav-list {
  list-style: none;
  display: flex;
}
.nav-item {
  margin-right: 15px;
}
.nav-link {
  color: #fff;
  text-decoration: none;
  font-weight: bold;
  transition: color 0.3s;
}
.social-icons {
  display: flex;
  align-items: center;
}
.social-icon {
  color: #fff;
  margin-left: 10px;
}
}

```



Рисунок 3.6 – Блок меню закладу

Вищевказаний код є лише прикладом і може відрізнятись від фактичного коду веб-сайту, який безпосередньо буде після тестування та оптимізації.

Верстка блоку меню здійснюється за допомогою списків ``. Кожен елемент меню складається з наступних елементів: зображення, назви, опису, ціни та можливості збільшити зображення.

У даному прикладі кожен елемент меню представлений як `` (пункт списку). Зображення страви розміщено всередині елемента `` за допомогою тегу ``, де `src` вказує на шлях до зображення, а `alt` містить опис страви для відображення у разі недоступності зображення.

Назва страви розміщена у тегу `<h3>`, опис страви - у тегу `<p>`, ціна - у тегу `` з класом `price`. Крім того, присутнє посилання `<a>` з класом `zoom`, яке дозволяє збільшити зображення страви.

Ви можете додати додаткові елементи меню, повторюючи структуру `` з відповідним вмістом. Ця верстка блоку меню дозволяє створити спискову структуру з зображеннями, назвами, описами та цінами страв, а також надає можливість збільшити зображення для отримання більш детальної інформації.

За допомогою *шаблонізатора Twig* легко створити декілька таких елементів меню шляхом повторення коду з використанням циклу або ітерації [17].

Ось приклад використання шаблонізатора Twig для створення декількох елементів меню:

```
<ul class="menu">
  {% for item in menuItems %}
    <li>
      
      <h3>{{ item.name }}</h3>
      <p>{{ item.description }}</p>
      <span class="price">Ціна: {{ item.price }}
грн</span>
    </li>
  {% endfor %}
</ul>
```

У цьому прикладі змінна `menuItems` представляє собою масив або колекцію

об'єктів, які містять дані для кожного елемента меню. За допомогою циклу `{% for %}` шаблон Twig проходить через кожен елемент `item` у `menuItems` і створює відповідний HTML-код для кожного елемента меню.

Цей підхід дозволяє легко повторювати блок коду для кожного елемента меню, що забезпечує швидку і просту генерацію багатьох елементів на основі даних зі змінної `menuItems`.

Фінальний вигляд дизайну головної сторінки сайту буде наступним:

1. Блок "Про нас":
 - Текстова інформація про нас розмічена в тегу `<p>`.
 - Контактні дані відображаються з використанням відповідних тегів (наприклад, `<address>` для адреси, `<a>` для посилань на електронну пошту або номер телефону).
 - Використовуються смайлики або інші візуальні елементи для підсилення емоційної складової.
2. Блок "Меню" або "Категорії":
 - Перераховані доступні категорії страв в закладі (наприклад, "Бургери", "Піци", "Закуски" тощо).
 - Елементи категорій є клікабельними, що дозволяє користувачам взаємодіяти з ними.
 - При наведенні курсору на елементи категорій вони збільшуються для наголошення можливості взаємодії з ними.
3. Блок "Галерея".
 - У цьому блоці будуть розміщені нові фото закладу, і користувачі матимуть можливість переглядати їх усіх у одному місці. Галерея автоматично перемикається між фотографіями, щоб користувачі могли насолоджуватись різними зображеннями закладу. Крім того, вона також має стрілки для ручного перемикання фотографій, що дозволяє користувачам контролювати процес перегляду. Код галереї буде наданий у додатках, щоб ви могли ознайомитись з його реалізацією.
4. Блок контактів і гугл-карти.

- У цьому блоку користувачі можуть наглядно ознайомитися з розташуванням закладу, шляхами до нього, графіком роботи та контактними даними. Всі посилання є клікабельними, що дозволяє користувачам одразу перейти до відповідної сторінки або здійснити дзвінок. Розмітка блоку контактів і гугл-карти відповідає правильним стандартам та забезпечує зручну взаємодію користувачів з цими елементами.

Завдяки цим блокам, головна сторінка сайту отримує привабливий вигляд, забезпечує зручну навігацію та інформативність для відвідувачів.

Дизайн головної сторінки буде адаптивним, що означає, що він буде пристосований до різних розмірів екрану, включаючи мобільні пристрої. Таким чином, блоки будуть адекватно відображатися на різних пристроях, забезпечуючи зручне користування сайтом незалежно від типу пристрою, на якому відкривається веб-сайт.

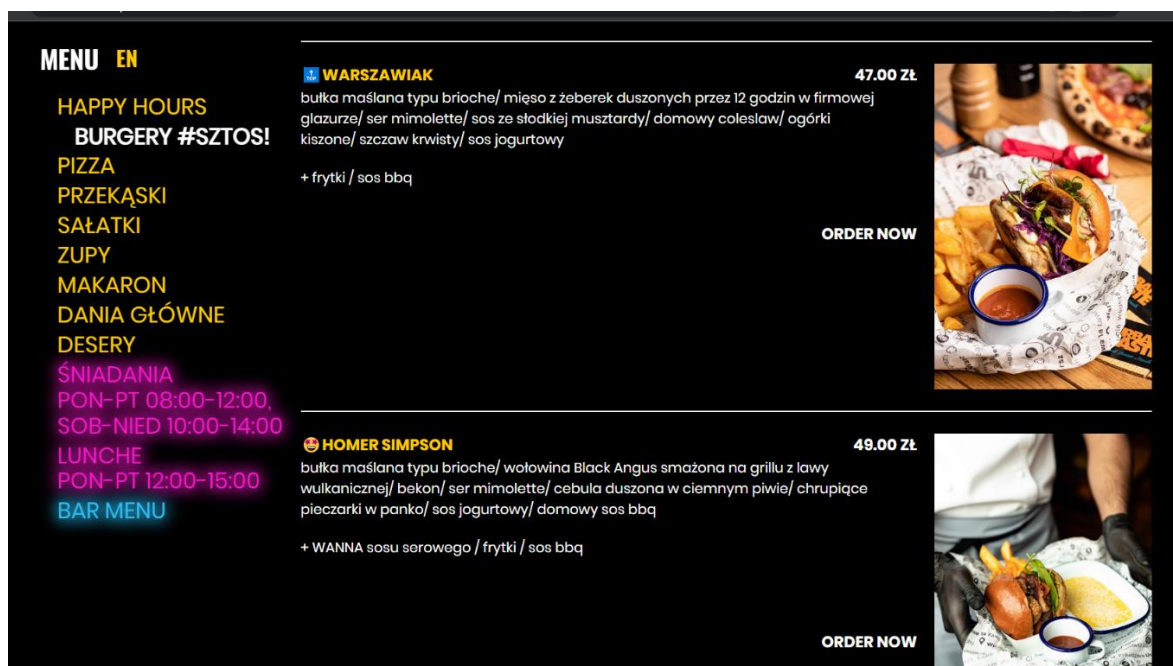


Рисунок 3.7 – Desktop сторінка меню закладу

Фінальний вигляд сторінки меню включає наступні елементи. Зліва розташовані категорії страв, які є клікабельними. При натисканні на категорію страви, сторінка автоматично прокручується до відповідного розділу меню, щоб користувачі зручно переглядали доступні страви.

Праворуч розміщений асортимент меню з детальним описом, цінами та іншою інформацією. Кожен елемент меню включає назву страви, опис, ціну та інші характеристики, що дозволяють користувачам зрозуміти, що пропонується в закладі. Цей блок надає повну інформацію про різноманітність страв, а також допомагає користувачам зробити свій вибір.

Такий розміщення категорій та асортименту меню забезпечує зручну навігацію та швидкий доступ до інформації про страви. Користувачі можуть легко орієнтуватись у меню та знайти потрібну страву з допомогою клікабельних категорій та зрозумілого відображення асортименту меню.

В результаті цього етапу мій веб-сайт отримує свою виглядову форму, де структура та зовнішній вигляд сторінок відповідають вимогам мого проекту. Використання HTML та CSS допомагає мені створити привабливий та професійний дизайн мого веб-сайту "Urban", який відповідає моїм завданням.

3.5 Тестування та оптимізація сайту

Під час оптимізації сайту були досягнуті максимальні показники ефективності, спеціальних можливостей та оптимальних методів, а також було звернуто увагу на оптимізацію для пошукових систем.

Ефективність сайту була максимально підвищена шляхом виконання різних оптимізаційних заходів, таких як мінімізація файлів CSS та JavaScript, стискання зображень, використання кешування та оптимізація запитів до бази даних. Це дозволило досягти швидкого завантаження сторінок та знизити час відклику.

Сайт був розроблений з урахуванням спеціальних можливостей, таких як адаптивний дизайн для різних пристроїв, покращений доступ для користувачів з особливими потребами та оптимізований контент для швидкого та зручного споживання.

Оптимальні методи розробки та програмування були застосовані для забезпечення ефективності та швидкості роботи сайту. Використання

компонентного підходу, шаблонізатора Twig та інших передових технологій дозволило створити чистий, легко зрозумілий та підтримуваний код.

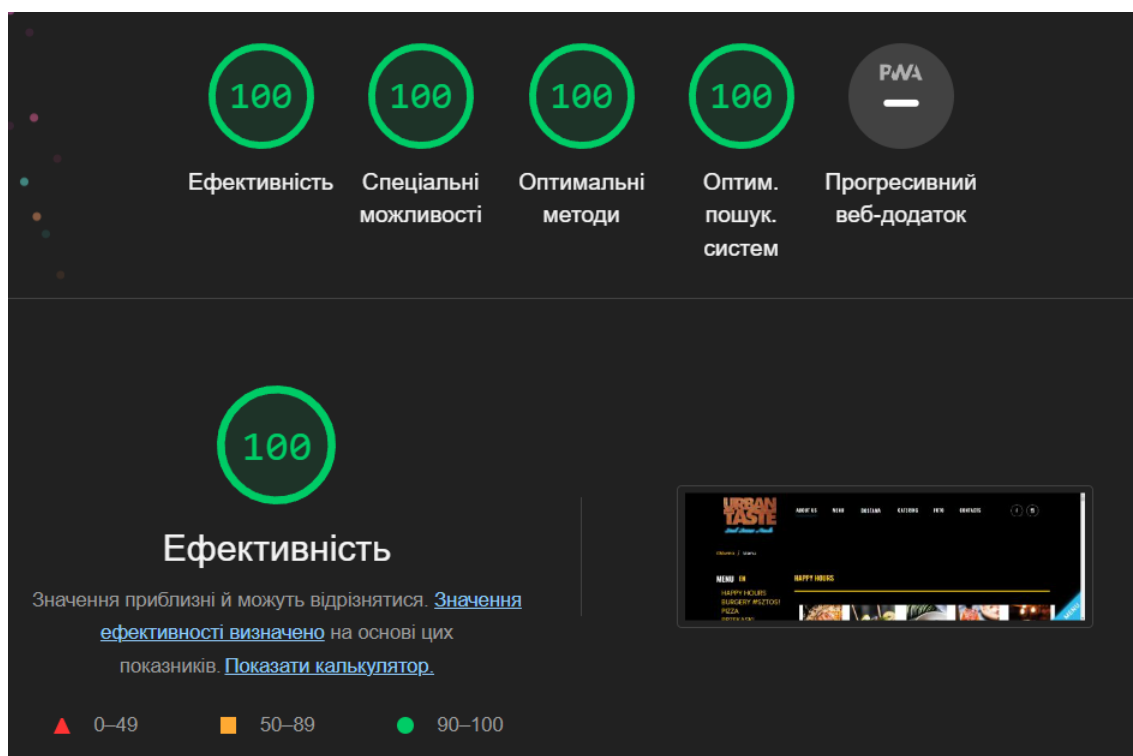


Рисунок 3.8 – Звіт з оптимізації сайту

Крім того, були вжиті заходи для оптимізації сайту для пошукових систем. Застосування семантичної верстки, правильного використання метатегів, а також оптимізація заголовків та ключових слів допомогли забезпечити кращу видимість сайту у пошукових системах та підвищити його пошукову позицію.

У результаті проведених оптимізаційних заходів сайт досяг максимальних показників ефективності, спеціальних можливостей, оптимальних методів та був оптимізований для пошукових систем, забезпечуючи високу продуктивність та зручний користувацький досвід.

3.6 Розгортання та підтримка

Процес заливки файлів проекту в систему управління контентом (CMS) включає кілька кроків. Спочатку необхідно завантажити всі файли сайту,

включаючи HTML-сторінки, зображення, стилі CSS та сценарії JavaScript, на сервер, на якому буде розміщений наш веб-сайт. Для цього можна використовувати FTP-клієнт або інше засіб передачі файлів.

Після завантаження файлів потрібно залогінитись в системі управління контентом, яку ми використовуємо, і створити новий проект або відкрити існуючий. Далі в системі управління контентом зазвичай є можливість завантажити файли з локального сервера. Ми обираємо завантаження всіх файлів, які ми перед тим завантажили на сервер.

Після завантаження файлів в систему управління контентом, можна розпочати їх організацію та налаштування. Це включає створення сторінок, налаштування меню, додавання зображень, описів та інших важливих даних для кожного розділу або елемента сайту.

Щодо покупки домену і хостингу, це можна зробити через різні сервіси, що пропонують реєстрацію доменних імен та послуги хостингу. Ми вибрали відповідні доменне ім'я для нашого сайту і зареєстрували його. Далі було сплачено хостинговий план, який надає необхідний простір на сервері для розміщення нашого сайту. Після оплати, налаштування хостингу зазвичай включає встановлення параметрів сервера, налаштування баз даних та інші необхідні кроки для правильної роботи сайту.

Залиття файлів проекту в систему управління контентом, налаштування сайту, а також покупка домену і хостингу є важливими етапами в процесі розгортання веб-сайту, оскільки це дозволяє нам створити присутність в Інтернеті та забезпечити доступ до нашого сайту користувачам.

Після завантаження файлів проекту в систему управління контентом і налаштування сайту, ми переходимо до етапу розгортання сайту. Цей процес включає розміщення нашого сайту на публічному сервері, щоб користувачі могли отримати до нього доступ через Інтернет. Для цього необхідно налаштувати веб-сервер, на якому буде розміщений наш сайт, налаштувати доменне ім'я, щоб відповідати IP-адресі сервера, і забезпечити необхідні налаштування безпеки.

Після успішного розгортання сайту ми переходимо до фази підтримки. Підтримка сайту включає регулярне оновлення контенту, виправлення помилок, захист від потенційних загроз безпеки, моніторинг роботи сайту і відповідь на запити користувачів. Ми також можемо використовувати аналітичні інструменти для відстеження відвідуваності сайту, аналізу популярних сторінок і функціональності, щоб постійно покращувати досвід користувачів.

Регулярні бекапи бази даних і файлів сайту також є важливою складовою підтримки. Це дозволяє відновити сайт у випадку втрати даних або несправності. Крім того, можуть проводитись оновлення системи управління контентом, тем і плагінів, щоб забезпечити безпеку і використовувати нові функції.

Важливо пам'ятати, що підтримка сайту є постійним процесом, який потребує уваги і ресурсів. Забезпечення безперебійної роботи, оновлення контенту, виправлення помилок і відповідь на запити користувачів є важливими аспектами, що допомагають забезпечити успіх нашого веб-проекту.

РОЗДІЛ 4. ІМПЛЕМЕНТАЦІЯ PHP-ШАБЛОНІЗАТОРА TWIG

4.1 Інструкція відвідувача

Розглянемо по крокам, як користуватись сайтом та на які блоки він розбитий. Після того, як ми заходимо на сайт, ми бачимо його головну сторінку. На початку сторінки відтворюється вражаюче відео, яке миттєво привертає мою увагу. Це відео створює початкове враження про сайт і заклад, а також передає атмосферу і стиль, які можна очікувати. Під відео розміщена важлива інформація про сам заклад, така як його розташування, контактні дані та години роботи.

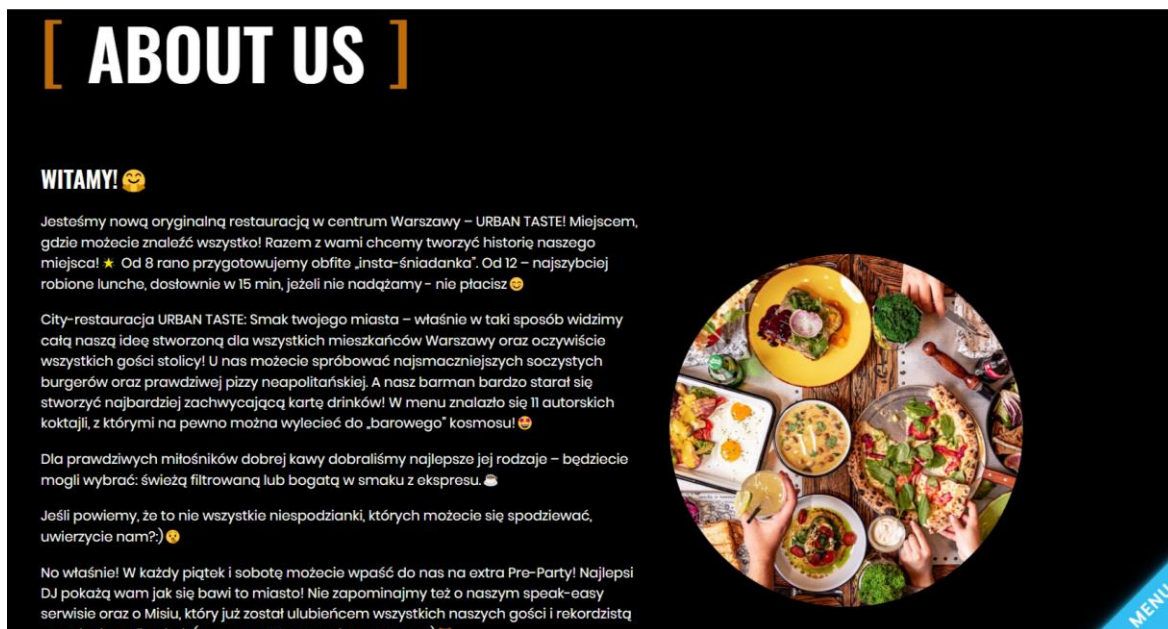


Рисунок 4.1 – Блок про нас на головній сторінці

Нижче, під цією інформацією, розташовані категорії їжі, які доступні у закладі. Ці категорії є клікабельними, що дозволяє мені безпосередньо перейти на сторінку Меню та зосередитись на виборі страви. Це дуже зручно, оскільки я одразу можу перейти до потрібної категорії і дізнатись більше про їжу, яка мене цікавить.

Усе це перше враження створює відчуття привітності та легкості взаємодії з сайтом, що заохочує мене досліджувати більше і ознайомлюватись з усіма можливостями закладу.



Рисунок 4.2 – Категорії страв

За допомогою зручної навігації та зрозумілого розташування блоків, я можу легко знайти потрібну інформацію і насолодитись своїм досвідом відвідування цього закладу.

При кліці на відповідну категорію з меню, я автоматично переходжу на сторінку Меню, де мені відображається весь асортимент страв з цієї категорії. Це дозволяє мені зосередитись на виборі страви саме з того розділу меню, який мене цікавить. На сторінці Меню я можу переглядати описи страв, їх склад, ціни та іншу важливу інформацію. Такий зручний перехід на сторінку Меню допомагає мені швидко знайти та обрати бажану страву і насолодитись смачними стравами у закладі.

Також, при кліці на картинку страви, я можу роздивитись її більш детально у повний розмір. Це дає мені можливість краще оцінити вигляд та привабливість страви перед замовленням. Крім того, на кнопку "Замовити", якщо натиснути на неї, з'являється модальне вікно з вибором сервісів для замовлення їжі. Це зручно, оскільки я можу обрати підходящий сервіс доставки або резервування столика безпосередньо з сайту. Такий функціонал спрощує процес замовлення їжі і робить його більш зручним для мене як відвідувача закладу.

Це найголовніша інформація що описує розробку кваліфікаційної роботи.

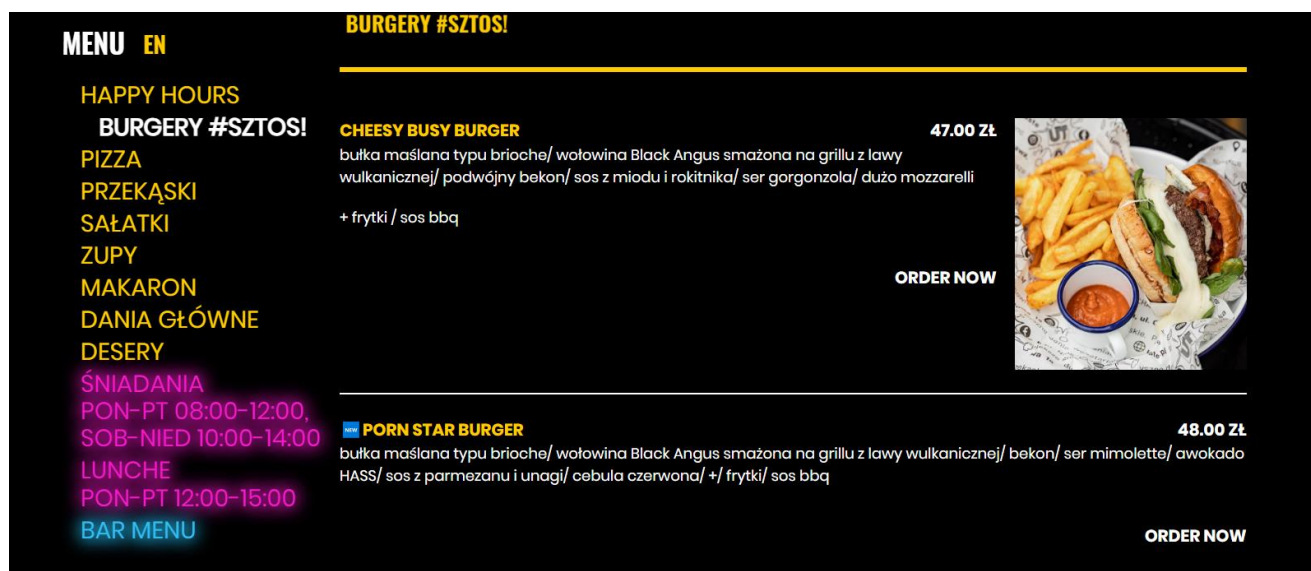


Рисунок 4.3 – Меню закладу

Решта розділів і блоків сайту були розроблені як доповнення і для надання стилістичного оформлення. Основна мета розробки сайту полягала в створенні зручного онлайн-меню. Вся функціональність і дизайн були спрямовані на те, щоб відвідувачі могли легко знайти і обрати страви з меню. Додаткові блоки, такі як галерея, контакти та інші, допомагали створити повний образ закладу і надавали додаткові можливості для взаємодії з відвідувачами.



Рисунок 4.4 – Вікно з сервісами доставки їжі

Додатково, відвідувачі закладу можуть зручно сканувати QR-код, який перенаправляє їх безпосередньо на сторінку меню сайту. Цей функціонал має декілька переваг:

- 1 Зручність і швидкість: Відвідувачам не потрібно шукати або

запитувати офіціанта про меню. Вони можуть миттєво отримати доступ до меню закладу просто сканувавши QR-код своїм смартфоном.

2 Актуальність і легка зміна: Меню на сайті може бути оновлене або змінене в реальному часі без необхідності друкувати нові фізичні версії меню. Відвідувачі завжди матимуть доступ до актуальної інформації про страви та їх ціни.

3 Деталізована інформація: Відвідувачі можуть отримати докладну інформацію про кожну страву, включаючи складники, опис, фотографії тощо. Це дозволяє їм зробити більш обґрунтований вибір і знайти страву, що відповідає їхнім потребам і смаковим вподобанням.

4 Екологічний аспект: Використання QR-кодів для доступу до меню допомагає зменшити використання паперу та зайвого друку. Це сприяє екологічній свідомості та відповідає сучасним тенденціям стосовно збереження навколишнього середовища.

Загалом, можливість сканування QR-коду для доступу до меню на сайті надає зручність, швидкість і актуальність інформації для відвідувачів закладу.

РОЗДІЛ 5. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1. Розробка логіко-імітаційної моделі виникнення травм і аварій

Методикою оцінки рівня небезпеки робочих місць, машин, виробничих процесів та окремих виробництв передбачено пошук об'єктивного критерію рівня небезпеки для конкретного об'єкта []. Таким показником вибрана ймовірність виникнення аварії, травми залежно від явища, що досліджується.

Для побудови логіко-імітаційної моделі процесу, формування і виникнення аварії та травми в процесі створення мікрокліматичних умов у приміщенні оцінюють відповідні небезпечні події. Кожній із них присвоєно ймовірність виникнення:

Шифр	Назва події	Ймовірність
P ₁	Відсутність захисного заземлення	0,02
P ₂	Пошкодження захисного заземлення	0,04
P ₃	Спрацювання складових захисту	0,1
P ₄	Неправильна експлуатація захисту	0,02
P ₅	Відсутність профілактичних заходів	0,2
P ₆	Відсутність захисного щита	0,12
P ₇	Недотримання правил вибору взуття	0,15
P ₈	Незнання правил техніки безпеки	0,1
P ₉	Відсутність засобів індивідуального захисту	0,2
P ₁₀	Легковажність	0,08

На основі наведених подій будемо матрицю логічних взаємозв'язків між окремими пунктами, графічна інтерпретація якої зображено на рис. 5.1.

Розрахуємо ймовірності виникнення подій, що формують логіко-імітаційну модель процесів створення мікрокліматичних умов. Розглянемо травмонебезпечну ситуацію, що виникає за умови роботи працівників із

електронезбезпекою.

Підставивши дані ймовірностей базових подій у формулу, отримаємо ймовірність події 13: $P_{13} = 0,2 + 0,4 - 0,2 \cdot 0,4 = 0,0592$.

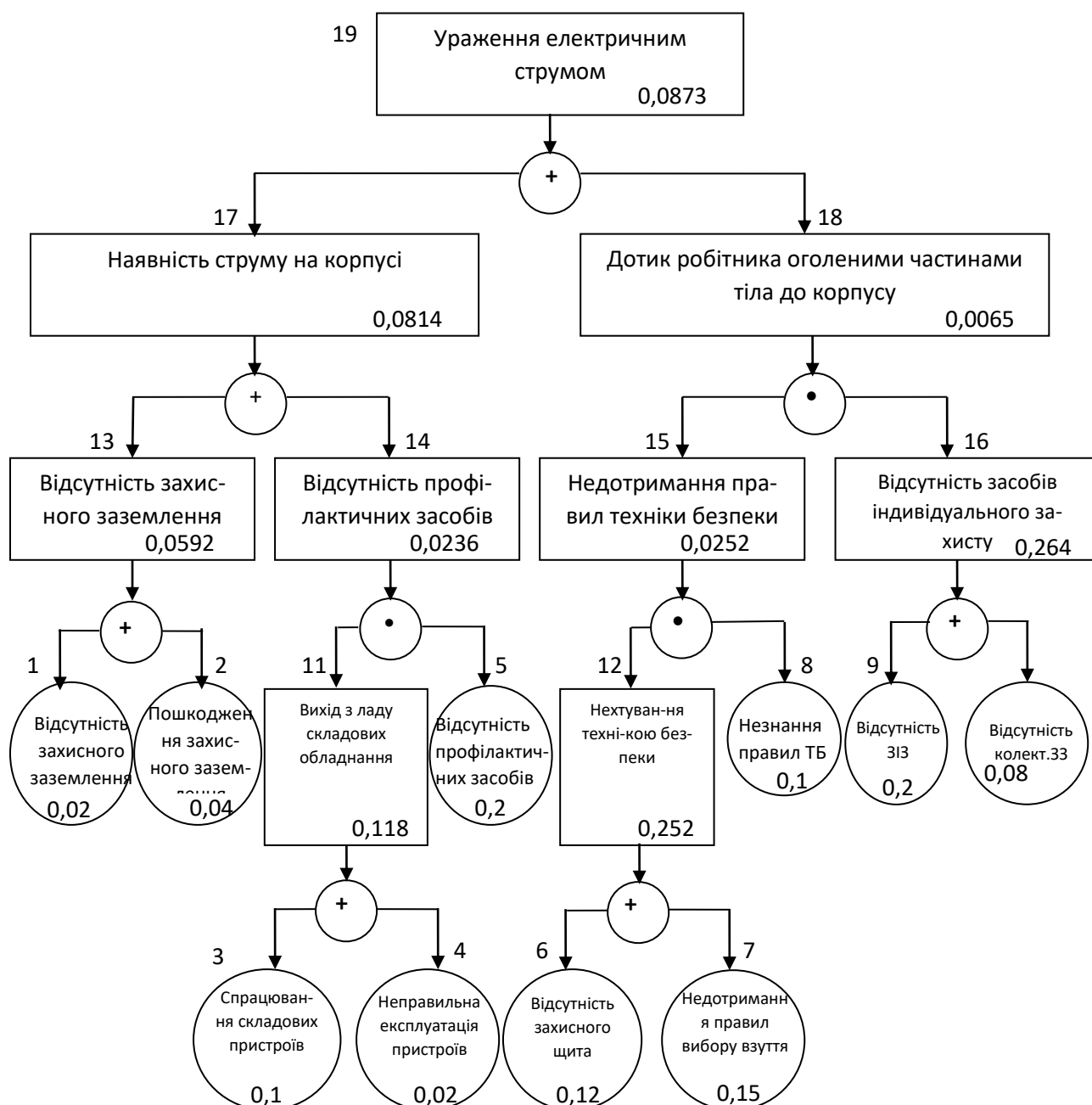


Рис. 5.1. Матриця логічних взаємозв'язків між окремими подіями травмонебезпечної ситуації []

Аналогічно визначаємо ймовірність інших подій:

$$P_{11} = P_4 + P_5 - P_4P_5 = 0,3 + 0,4 - 0,3 \cdot 0,4 = 0,118.$$

$$P_{12} = P_6 + P_7 - P_6P_7 = 0,3 + 0,5 - 0,3 \cdot 0,5 = 0,252.$$

$$P_{16} = P_9 + P_{10} - P_9P_{10} = 0,2 + 0,15 - 0,2 \cdot 0,15 = 0,264.$$

$$P_{14} = P_{11} \cdot P_5 = 0,118 \cdot 0,2 = 0,0236.$$

$$P_{15} = P_{12} \cdot P_8 = 0,252 \cdot 0,1 = 0,0252.$$

$$P_{17} = P_{13} + P_{14} - P_{13} \cdot P_{14} = 0,592 + 0,0236 - 0,0592 \cdot 0,0236 = 0,0814.$$

$$P_{18} = P_{15} \cdot P_{16} = 0,264 \cdot 0,0252 = 0,0065.$$

$$P_{19} = P_{17} + P_{18} - P_{17} \cdot P_{18} = 0,0065 + 0,0814 - 0,0065 \cdot 0,0814 = 0,0873.$$

Таким чином, ймовірність перекидання машини та наслідкового виникнення травми працівника є досить мала і становить – $P_{19} = 0,0873$.

5.2. Планування заходів із покращення умов праці

До заходів щодо покращення умов праці належать всі види діяльності, спрямовані на попередження, нейтралізацію або зменшення негативної дії шкідливих і небезпечних виробничих факторів на працівників.

Рівень умов праці оцінюють порівнянням за фактичними і нормативними значеннями узагальнених (групових) показників.

Заходи щодо поліпшення умов праці здійснюють з метою створення безпечних умов праці шляхом:

- доведення до нормативного рівня показників виробничого середовища за елементами умов праці;
- захисту працівників від дії небезпечних і шкідливих виробничих факторів.

До показників ефективності заходів щодо поліпшення умов праці належать:

- а) зміни стану умов праці:
 - зміна кількості засобів виробництва, приведених у відповідність до вимог стандартів безпеки праці;
 - покращання санітарно-гігієнічних показників;
 - покращання психофізичних показників, зменшення фізичних і нервово-психічних навантажень, в т.ч. монотонних умов праці;
- б) соціальні результати заходів:
 - збільшення кількості робочих місць, що відповідають нормативним

вимогам;

- зниження рівня виробничого травматизму;
- престиж та задоволення працею.

Отже, на покращення охорони праці потрібно виділити кошти на відновлення вентиляційних систем у ремонтних майстернях, естетично оформити приміщення офісу, відновити кабінет з охорони праці, поновити протипожежний інвентар.

5.3. Безпека в надзвичайних ситуаціях

Актуальність проблеми природно-техногенної безпеки для населення і території, зумовлена зростанням втрат людей, що спричиняється небезпечними природними явищами, промисловими аваріями та катастрофами []. У системі цивільної оборони окремого господарства необхідно забезпечити захист населення таким чином:

Укриття в захисних спорудах, якому підлягає усе населення відповідно до приналежності, досягається створенням фонду захисних споруд.

Евакуаційні заходи, які проводяться в містах та інших населених пунктах, які мають об'єкти підвищеної небезпеки, а також у воєнний час, основним способом захисту населення є евакуація і розміщення його у позаміській зоні.

Медичний захист проводиться для зменшення ступеня ураження людей, своєчасного надання допомоги постраждалим та їх лікування, забезпечення епідеміологічного благополуччя в районах надзвичайних ситуацій.

Радіаційний і хімічний захист включає заходи щодо виявлення і оцінки радіаційної та хімічної обстановки, організацію і здійснення дозиметричного та хімічного контролю, розроблення типових режимів радіаційного захисту, забезпечення засобами індивідуального захисту, організацію і проведення спеціальної обробки.

ВИСНОВКИ

В даних дослідження Інтернет-аудиторії України gemiusAudience з 1 червня 2023 року здійснено переоцінку популяції у зв'язку з війною в Україні. Загальна Інтернет-аудиторія України на всіх платформах в червні 2023 року становить 25,6 млн Real Users (зменшилась на 2,7 млн Real Users). Загальна аудиторія на ПК становить 14,2 млн Real Users (зменшилась на 5,4 млн Real Users), на Телефонах – 23,7 млн Real Users (зменшилась на 1,7 млн Real Users), на Планшетах – 2,9 млн Real Users.

Багато сайтів на якийсь спосіб займаються заробітком, через розміщення реклами або надання платних послуг користувачам. Однак, вони були створені спочатку не для прибутку, а для задоволення потреб користувачів, тому їх можна вважати некомерційними. Але в таких випадках границя між комерційними і некомерційними сайтами стає дещо нечіткою, і багато з цих сайтів можна назвати "умовно некомерційними".

Виконаний аналіз галузі та ІТ-систем надання послуг дав змогу визначити ключові фактори та поточні вимоги до розробки. Проведена комплексна оцінка конкурентів та існуючих альтернатив, дала змогу визначити їхні сильні та слабкі сторони.

В основу вирішення поставлених завдань кваліфікаційної системи покладено унікальні характеристики предметної області та оптимізовано веб-інтерфейс і навігацію сайту, щоб забезпечити безперебійну роботу користувачів. В результаті розроблено сайт закладу, який дозволяє користувачам отримати доступ до інформації про послуги та товар, включаючи контактні дані, місцезнаходження, години роботи, опис ресторану, меню, ціни на страви, а також можливість оформити замовлення через службу доставки.

Для досягнення цілей кваліфікаційної роботи використано мову розмітки гіпертексту HTML. Це дозволило створити структуровану основу для сайту, окрім того, CSS (каскадні таблиці стилів) допомогли нам створити привабливий та естетичний дизайн, що покращує загальний візуальний вигляд сайту.

Один із головних інструментів, який використано для реалізації функціональності сайту – мова програмування JavaScript. Завдяки широкому спектру функціональних можливостей, які надає JS, створено інтерактивні елементи на сайті, такі як форми замовлення та функціонал служби доставки. Це значно полегшило користувачам взаємодію з сайтом та зробило процес замовлення зручним та ефективним.

Розробка веб-сайту велася із використанням редактора коду VS Code, який надав нам зручне та продуктивне середовище для програмування. Також використано систему збірки контенту Gulp, щоб автоматизувати деякі завдання розробки та покращити продуктивність роботи.

Однак, не менш важливим інструментом, що сприяв розробці, став шаблонізатор Twig. Використання Twig дозволило ефективно створювати шаблони для веб-сторінок та розбити їх на компоненти, що значно спростило процес розробки та підтримки сайту. Крім того, Twig надав можливість зручно маніпулювати даними та розробляти динамічні сторінки з урахуванням специфічних потреб закладу.

У результаті роботи створено сайт закладу, який успішно виконує всі поставлені завдання. Користувачі мають можливість з легкістю отримати доступ до актуальної інформації, знайти необхідну контактну інформацію, ознайомитись із розташуванням та годинами роботи. Крім того, вони можуть ознайомитись із різноманітним меню та цінами на страви, а також зручно оформити замовлення через службу доставки.

БІБЛІОГРАФІЧНИЙ СПИСОК

1. Вандер Воф, М. Responsive. Web Design. A Book Apart, 2011. 150 с.
2. Веб студія WebTune. Види сайтів та їх функціонал. URL: <https://webtune.com.ua/statti/web-rozrobka/vydy-sajtiv-ta-yih-funkczional/>
3. Вікіпедія. JavaScript. URL: <https://uk.wikipedia.org/wiki/JavaScript>
4. Веб-документація MDN. JavaScript. URL: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript
5. Використання шаблонів Twig у веб-розробці: ефективність та зручність / під заг. ред. О. М. Соколової. Одеса: Видавництво "WebDev", 2022. 160 с.
6. Винокурова Л.Е., Васильчук М.В., Гаман М.В. Основи охорони праці: Підручник для проф.-техн. навч. закладів, 2001, 192с.
7. Вікіпедія. Редактор початкового коду. URL: https://uk.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80_%D0%BF%D0%BE%D1%87%D0%B0%D1%82%D0%BA%D0%BE%D0%B2%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BE%D0%B4%D1%83
8. Вікіпедія. Безпека життєдіяльності. URL: https://uk.wikipedia.org/wiki/%D0%91%D0%B5%D0%B7%D0%BF%D0%B5%D0%BA%D0%B0_%D0%B6%D0%B8%D1%82%D1%82%D1%94%D0%B4%D1%96%D1%8F%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%96
9. Даддаріо Дж. HTML5 і CSS3 в реальному світі. SitePoint, 2011. 390 с.
10. Дакетт Д. HTML та CSS, Ексмо, 2021, 480 с.
11. Джонсон Е. Розробка веб-сайтів з використанням Node.js, Express та MongoDB. O'Reilly Media, 2014. 352 с.
12. Діаграми прецедентів. URL: <https://evergreens.com.ua/ua/articles>

13. Діаграми класів. URL: <https://www.ua5.org/oop/392-diagrami-klasiv.html>
14. Ерік Ф., Робсон Е. Head First. Програмування JavaScript, 2022, 672 с.
15. Макфарланд Д. CSS: Практичний посібник. O'Reilly Media, 2017. 840с.
16. Майер Е.А. CSS: Визначний посібник: Візуальна презентація для вебу. O'Reilly Media, 2017. 1080 с.
17. Новіков Ф.В., Богатов О.І., Навчальний посібник «Безпека життєдіяльності», Харків 2018, 250 с.
18. Програмування на JavaScript для веб-розробників: практичні аспекти та приклади / за ред. В. П. Кравченка. Львів: 2022. 220 с.
19. Робін Н. Створюємо динамічні веб-сайти за допомогою PHP, MySQL, JavaScript, CSS та HTML5. 6-те вид, Прес, 2019, 816 с.
20. Розробка веб-сайту з використанням HTML: основи та принципи / під заг. ред. О. В. Петренка. Київ: Видавництво "WebCode", 2020. 200 с.
21. Фленаган Д. JavaScript. Докладний посібник. Київ: Науковий Світ, 2023. 722 с.
22. Фріман Е., Робсон, Е. Head First HTML і CSS: Посібник для самостійного навчання. O'Reilly Media, 2012. 768 с.
23. Ширіш Н. Навчання веб-дизайну: Посібник для початківців з HTML, CSS, JavaScript та веб-графікою. O'Reilly Media, 2018. 592 с.
24. CSS у веб-розробці: стилізація та оформлення веб-сайтів / за заг. ред. І. І. Мельникової; уклад. М. Ю. Ковальова. Харків: 2021. 180 с.
25. DOU. Рейтинг мов програмування 2023. URL: <https://dou.ua/lenta/articles/language-rating-2023/>
26. JavaScript. URL: <https://learn.javascript.ru/intro>
27. HTML. URL: <https://developer.mozilla.org/ru/docs>
28. PHP 8. URL: <https://www.php.net/releases/8.0/ru.php>
29. PHPStorm. URL: <https://itpro.ua/product/jetbrains>
30. SEO-оптимізація веб-сайту: стратегії та методи / за заг. ред. Н. О. Данилова ; уклад. І. М. Гриньова. Київ: Видавництво "SEO-Code", 2023. 190 с.

31. TWIG. Компілюючий обробник шаблонів. URL:
<https://twig.symfony.com/>
32. Laravel Routing. URL: <https://laravel.com/docs/10.x/routing>
33. Laravel Middleware. URL: <https://laravel.com/docs/10.x/middleware>
34. Laravel HTTP Request. URL: <https://laravel.com/docs/10.x/requests>
35. HTTP Session. URL: <https://laravel.com/docs/10.x/session>
36. Error Handling Laravel. URL: <https://laravel.com/docs/10.x/errors>

ДОДАТКИ

Додаток А.

Фрагмент коду сторінки меню

```

var toc_link = document.querySelectorAll(".js-products-nav");
var toc_list = document.querySelector(".js-products-list");
if (toc_list) {
    var all_id_list = toc_list.querySelectorAll("a[href*=toc-]");
}
var toc_title = document.querySelectorAll(".c-products__right
*[id^=toc]");
var window_height = window.innerHeight;
var top_marign = 0.01;
var bottom_margin = 0.71;
if (toc_list) {
    function toc() {
        [].forEach.call(toc_title, function (e) {
            var target_bound = e.getBoundingClientRect();
            let target_bound_id = e.id.slice(4);
            let index = 0;
            if( target_bound.bottom > window_height * top_marign &&
target_bound.top < window_height * ( 1 - bottom_margin ) ) {
                for (var i = 0; i < toc_link.length; i++) {
                    toc_link[i].classList.remove('is-active');
                    if (toc_link[i].getAttribute('href').slice(5) ==
target_bound_id)
                        index = i;
                }
                toc_link[index].classList.add('is-active');
            }
        })
    }
    if (window.matchMedia("(min-width: 768px)").matches){
        toc();
        [].forEach.call(all_id_list, function(e) {
            e.addEventListener("click", function() {
                setTimeout(function () {
                    window.scrollTo(0, -25);
                    for (var i = 0; i < toc_link.length; i++) {
                        toc_link[i].classList.remove('is-active');
                    }
                    e.classList.add('is-active');
                }, 1)
            });
        });
    }
}
}

```

```

var frames_count = 20;
var check_is_scrolling = false;

var all_menu_list = document.querySelectorAll('a[href^="#go-
"]');
var all_menu_arr = [].slice.call(all_menu_list);
if (all_id_list) {
    var all_id_arr = [].slice.call(all_id_list);
    var all_anchor = all_id_arr.concat(all_menu_arr);
}
else{
    var all_anchor = all_menu_arr;
}

all_anchor.forEach( function (item) {
    item.addEventListener('click', function(e) {
        e.preventDefault();

        var element_with_id =
document.querySelector(item.getAttribute('href'));
        var coord_y =
element_with_id.getBoundingClientRect().top;
        var animation_speed = Math.abs(coord_y / 200);

        var previos_cord;
        var current_cord = coord_y;
        var pole = current_cord >= 0 ? true : false;

        if (!check_is_scrolling) {
            check_is_scrolling = true;

            let scroller = setInterval(function() {
                let scrollBy = coord_y / frames_count;

                if (element_with_id.getBoundingClientRect().top >
100 || element_with_id.getBoundingClientRect().top < -100) {
                    previos_cord =
element_with_id.getBoundingClientRect().top;
                    window.scrollBy(0, scrollBy);
                    current_cord =
element_with_id.getBoundingClientRect().top;

                    if (pole && current_cord < 0){
                        coord_y =
document.querySelector(item.getAttribute('href')).getBoundingClientRe
ct().top + window.pageYOffset;

```

```

        setTimeout(() => window.scrollTo(0,
coord_y), 5);
        clearInterval(scroller);
        check_is_scrolling = false;
    }
    else if (!pole && current_cord > 0) {
        coord_y =
document.querySelector(item.getAttribute('href')).getBoundingClientRect().top + window.pageYOffset;
        setTimeout(() => window.scrollTo(0,
coord_y), 5);
        clearInterval(scroller);
        check_is_scrolling = false;
    }
    else if (previos_cord ==
element_with_id.getBoundingClientRect().top) {
        coord_y =
document.querySelector(item.getAttribute('href')).getBoundingClientRect().top + window.pageYOffset;
        setTimeout(() => window.scrollTo(0,
coord_y), 5);
        clearInterval(scroller);
        check_is_scrolling = false;
    }
    } else {
        coord_y =
document.querySelector(item.getAttribute('href')).getBoundingClientRect().top + window.pageYOffset;
        setTimeout(() => window.scrollTo(0, coord_y),
5);
        clearInterval(scroller);
        check_is_scrolling = false;
    }
    }, animation_speed);
    }
    });
});

```