

**МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ**

**ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І
СИСТЕМ**

КВАЛІФІКАЦІЙНА РОБОТА

**на тему: РОЗРОБКА ВЕБДИЗАЙНУ ТА СТВОРЕННЯ ВЕБСАЙТУ
РЕСТОРАННОГО БІЗНЕСУ**

Здобувача вищої освіти першого (бакалаврського) рівня,
IV курсу, групи ЦДБ-21-1, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Технології цифрового дизайну»


(підпис)

Стрільця Дмитра Андрійовича
(прізвище, ім'я, по батькові)

Науковий керівник


(підпис)

Гладченко Оксана Вікторівна,
к.пед.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь,
вчене звання)

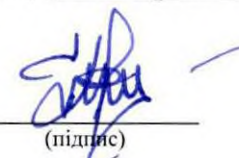
Керівник освітньо-
професійної програми


(підпис)

Ратушняк Тетяна Володимирівна,
к.ф.-м.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь,
вчене звання)

Допущено до захисту за рішенням кафедри комп'ютерних та інформаційних
технологій і систем протокол від «13» травня 2025 № 18

Завідувач кафедри


(підпис)

Філоненко Михайло
Миколайович, к.ф.-м.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь,
вчене звання)

Реєстрація № 80

29.05.2025

(дата)



(підпис відповідальної особи)

Результати перевірки

90 захисту
(до захисту, на доопрацювання)

29.05.2025

(дата)

Трух -

(підпис)

Результат захисту

4 (добре)
(оцінка)

11.06.2025

(дата)



(підпис)

Голова комісії:



(підпис)

В.М. Франчук

Члени комісії:



(підпис)

М.М. Філоненко



(підпис)

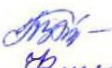
Т.В. Ратушняк



(підпис)

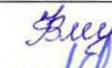
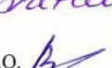
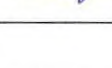
В.О. Ніжегородцев

БЛАНК ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії, секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	від 0 до 30	25		Гладченко О.В.  Франчук В.М.  Філоненко М.М.  Ратушняк Т.В.  Ніжегородцев В.О.  Суховерхівка Л.П. 
2. Дотримання вимог щодо оформлення	від 0 до 10	8		
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	4		
4. Рецензент	від 0 до 5	3		
РАЗОМ БАЛІВ	від 0 до 50	40		

Оцінювання кваліфікаційної роботи науковим керівником – 42 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підпис)
82	B	4 (добре)	11.06.2025	Франчук В.М.  Філоненко М.М.  Ратушняк Т.В.  Ніжегородцев В.О.  Суховерхівка Л.П. 

ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	
60-69	D	
50-59	E	3 (задовільно)
35-49	FX	2 (незадовільно)

Факультет фінансів та цифрових технологій
Кафедра комп'ютерних та інформаційних технологій і систем
Галузь знань: 12 «Інформаційні технології»
Спеціальність: 122 «Комп'ютерні науки»
Освітня програма: «Технології цифрового дизайну»

Затверджую

Завідувач кафедри

«14» листопада 2024

Філоненко М.М.

Завдання на кваліфікаційну роботу
здобувачу вищої освіти

Стрільцю Дмитру Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи Розробка вебдизайну та створення вебсайту ресторанного бізнесу

Затверджена наказом ректора від «13» листопада 2024 № 1717

2. Строк здачі закінченої роботи «12» травня 2025

3. Мета роботи розробка сучасного односторінкового вебсайту “Strelik” для інформування клієнтів, онлайн-бронювання столиків і замовлення страв.

Об'єкт дослідження створення вебсайту ресторанного бізнесу: інформування, бронювання, замовлення і взаємодія користувача.

Предмет дослідження методи, інструменти та технології UX/UI-дизайну й реалізації динамічного контенту.

Орієнтовна база дослідження прикладні матеріали сайтів-конкурентів (“Пузата Хата”, “Львівські Круасани”, “Сушія”), MDN WebDocs.

4. Вихідні дані до роботи визначаються на основі теперішніх прикладів ресторанних вебсайтів, вимоги до UX/UI дизайну, адаптивності, сучасні технології веброзробки (HTML, CSS, JavaScript), вимоги до зручності для користувача та швидкодії.

5. Перелік графічного матеріалу знімки з екрана основних інтерфейсних елементів вебсайту: головна сторінка, секція “Меню”, форма бронювання, кошик замовлення, мобільна версія навігації, блок “Контакти”, адаптивні елементи сайту та фрагменти коду.

6. Зміст кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП. РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ СТВОРЕННЯ ВЕБ-САЙТІВ. 1.1.

Історія вебтехнологій. 1.2. Виникнення комп'ютерних мереж і перший вебсайт.

1.3. Розвиток HTML та JavaScript. РОЗДІЛ 2. ВЕБСАЙТИ РЕСТОРАННОГО

БІЗНЕСУ ТА ЇХ ОСОБЛИВОСТІ. 2.1. Ціль ресторанних сайтів та їх типи. 2.2

Аналіз існуючих рішень. 2.3 Вимоги до вебсайту, технічні та функціональні.

РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА СТВОРЕННЯ ВЕБСАЙТУ. 3.1.

Прототипування та дизайн у Figma. 3.2. Структура сайту. 3.3. Вибір технологій

для розробки. 3.4. Функціональність вебсайту. ВИСНОВКИ. СПИСОК

ВИКОРИСТАНИХ ДЖЕРЕЛ.

7. Календарний план виконання роботи

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	Фактично
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	До 30.10.2024	До 30.10.2024
2	I розділ (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	12.03.2025	12.03.2025
3	II розділ (аналіз експериментальної частини про методики досліджень)	18.03.2025	18.03.2025
4	III розділ (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	21.04.2025	21.04.2025
5	Висновки	15.05.2025	15.05.2025
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	18.05.2025	18.05.2025
7	Отримання відгуку керівника	03.06.2025	03.06.2025
8	Отримання зовнішньої рецензії	05.06.2025	05.06.2025

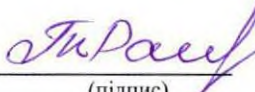
8. Дата видачі завдання «14» листопада 2024

9. Керівник кваліфікаційної роботи


(підпис)

Гладченко О.І.
(прізвище, ініціали)

10. Керівник освітньо-професійної програми


(підпис)

Ратушняк Т.В.
(прізвище, ініціали)

11. Завдання до виконання отримав здобувач вищої освіти


(підпис)

Стрілець Д.А.
(прізвище, ініціали)

АНОТАЦІЯ

Розробка вебдизайну та створення вебсайту ресторанного бізнесу

Мета роботи полягає в розробці сучасного односторінкового вебсайту “Strelik” для інформування клієнтів, онлайн-бронювання столиків і замовлення страв.

Завданнями роботи є:

- Огляд історії та сучасних веб-технологій;
- Аналіз сайтів-конкурентів;
- Прототипування в Figma;
- Верстка адаптивного інтерфейсу на HTML/CSS;
- Впровадження фільтрації меню й обробки форм на JavaScript.

Предметом дослідження є методи, інструменти та технології UX/UI-дизайну й реалізації динамічного контенту.

Об'єктом дослідження є процеси створення вебсайту ресторанного бізнесу: інформування, бронювання, замовлення і взаємодія користувача.

Базою дослідження є прикладні матеріали сайтів-конкурентів (“Пузата Хата”, “Львівські Круасани”, “Сушія”), MDN WebDocs.

За результатами дослідження сформульовані рекомендації щодо структури та функціональності ресторанного сайту й продемонстрована реалізація односторінкового лендінгу.

Одержані результати можуть бути застосовані при створенні вебресурсів для закладів ресторанного бізнесу.

Кваліфікаційна робота містить 46 сторінок, 39 рисунків, список використаних джерел з 43 найменувань.

Ключові слова: вебдизайн; ресторанний бізнес; адаптивність; UX/UI; JavaScript; динамічний контент.

Рік виконання кваліфікаційної роботи: 2025.

Рік захисту: 2025.

ABSTRACT

Development of Web Design and Creation of a Restaurant Business Website

Objective of the Work: To develop a modern single-page website, “Strelik” for informing customers, facilitating online table reservations, and enabling meal orders.

Tasks:

- Review the history and current state of web technologies;
- Analyze competitor websites;
- Prototype the interface in Figma;
- Coding of a responsive interface in HTML/CSS;
- Implementation of menu filtering and form processing in JavaScript.

Subject of Research: Methods, tools, and technologies of UX/UI design and dynamic content implementation.

Object of Research: The processes involved in creating a restaurant-business website: information delivery, booking, ordering, and user interaction

Research Base: Applied materials from competitor sites (“Puzata Hata,” “Lviv Croissants,” “Sushiya”) and MDN WebDocs.

Results and Recommendations: Recommendations have been formulated regarding the structure and functionality of the restaurant website, and the implementation of a single-page landing has been demonstrated.

Practical Application: The results obtained can be applied when creating web resources for establishments in the restaurant business.

Qualification Paper Contains: 46 pages, 39 figures, and a reference list of 43 sources.

Keywords: web design; restaurant business; responsiveness; UX/UI; JavaScript; dynamic content.

Year of Completion: 2025.

Year of Defense: 2025.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ.....	3
ВСТУП	4
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ СТВОРЕННЯ ВЕБСАЙТІВ	7
1.1. Історія вебтехнологій.....	7
1.2. Виникнення комп'ютерних мереж і ARPANET – перший у світі вебсайт. 9	
РОЗДІЛ 2. ВЕБСАЙТИ РЕСТОРАННОГО БІЗНЕСУ ТА ЇХ ОСОБЛИВОСТІ .	13
2.1. Ціль ресторанних сайтів та їх типи	13
2.2. Аналіз наявних рішень	15
2.3. Вимоги до вебсайту, технічні та функціональні.....	19
РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА СТВОРЕННЯ ВЕБСАЙТУ	22
3.1. Прототипування та дизайн у Figma.....	22
3.2. Структура сайту	23
3.3. Вибір технологій для розробки	33
3.4. Функціональність вебсайту.....	34
ВИСНОВКИ.....	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	44

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

AJAX – Asynchronous JavaScript and XML (технологія асинхронного обміну даними без перезавантаження сторінки).

ARPANET – Advanced Research Projects Agency Network (перша пакетно-комутована мережа).

CSS – Cascading Style Sheets (каскадні таблиці стилів).

DTD – Document Type Definition (визначення типу документа).

ENQUIRE – Early ENQUIry into Information RetRIEval (рання система управління інформацією, попередник WWW).

HTML – HyperText Markup Language (мова розмітки гіпертексту).

HTTP – HyperText Transfer Protocol (протокол передавання гіпертексту).

IP – Internet Protocol (інтернет-протокол).

ISO – International Organization for Standardization (Міжнародна організація зі стандартизації).

JS – JavaScript (мова програмування для створення динаміки на веб-сторінках).

SGML – Standard Generalized Markup Language (стандартна узагальнена мова розмітки).

SPA – Single Page Application (односторінковий веб-додаток).

TCP – Transmission Control Protocol (протокол керування передачею).

UI – User Interface (інтерфейс користувача).

URL – Uniform Resource Locator (уніфікатор ресурсу, адреса в інтернеті).

UX – User Experience (досвід користувача).

WWW – World Wide Web (Всесвітня павутина).

ВСТУП

Актуальність теми. У сучасних умовах розвитку ресторанного бізнесу цифрове представництво закладу відіграє ключову роль у формуванні першого враження та залученні клієнтів. Односторінкові вебсайти, що забезпечують швидкий доступ до інформації про меню, онлайн бронювання столиків і замовлення страв, дозволяють підвищити конверсію відвідувачів у реальних гостей закладу. Проте масові шаблонні рішення не враховують потреби адаптивного UX/UI дизайну та продуктивності, що призводить до втрати потенційної аудиторії. Розробка персоналізованого лендінгу “Strelik” є актуальною для забезпечення високої зручності користування (юзабіліті), та залучення клієнтів.

Дослідження у сфері вебдизайну для ресторанів зазвичай зосереджуються окремо на UX/UI принципах, адаптивній верстці або інтеграції форм бронювання. Проте комплексного підходу, який об’єднує аналіз конкурентного середовища, прототипування, реалізацію адаптивного інтерфейсу та динамічну фільтрацію меню, досі бракує.

Стан і ступінь проблеми в спеціальній літературі. В Україні теоретичні та практичні аспекти вебдизайну і створення вебсайтів висвітлено в роботах низки вітчизняних дослідників. Зокрема, Винарева О. В. (2005) у дисертації “Структурні, семантичні й прагматичні аспекти англomовних торгівельних назв в Internet” проаналізувала мовні й графічні компоненти вебсайтів. Чекмарьова Л. О. (2008) досліджувала організацію документно-інформаційних систем на державних вебресурсах, а Шукало І. М. (2008) розглянула семантичні й комунікативні аспекти рекламних інтерфейсів британських туристичних компаній. Бородаєв Д. В. (2004) у праці “Вебсайт як об’єкт графічного дизайну” визначив базові принципи візуального оформлення сторінок. Науковцем Кушнір В. В. (2024) досліджено розвиток підприємницької компетентності майбутніх веб дизайнерів у професійній підготовці. Крім того, питаннями вебдизайну та вебсайтів в навчально-методичному аспекті розглянуто

Пасічник О. Г., Романюк О. Н., Угрин Д. І., Куліш Л. А., Огурцов В. В., Трофименко О. Г., Цеслів О. В., Поморцева О. Є. та ін.

Закордонні фахівці Тім Бернерс-Лі (Tim Berners-Lee, 1989) у пропозиції “Information Management: A Proposal” заклав основу гіпертекстових мереж і сучасних стандартів HTML; Якоб Нільсен (Jakob Nielsen, 1993) у монографії “Usability Engineering” розвинув методики юзабіліті-тестування; Дон Норман (Don Norman, 1988) у книзі “The Design of Everyday Things” сформулював принципи дизайну, орієнтованого на користувача; Стів Круг (Steve Krug, 2000) у праці “Don’t Make Me Think” обґрунтував концепцію інтуїтивної навігації вебресурсами; Люк Вроблевскі (Luke Wroblewski, 2011) в “Mobile First” звертав увагу на пріоритеті мобільного UX та ін.

Попри різнопланові напрацювання як українських, так і закордонних дослідників, не існує єдиного, інтегрованого підходу до розробки односторінкових лендінгів (one-page site), який би одночасно поєднував сучасні UX/UI-принципи, адаптивну верстку та динамічну фільтрацію контенту.

Мета дослідження полягає в розробці сучасного односторінкового вебсайту “Strelik” для інформування клієнтів, онлайн-бронювання столиків і замовлення страв.

Завданнями дослідження є:

- Огляд історії та сучасних веб-технологій;
- Аналіз сайтів-конкурентів;
- Прототипування в Figma;
- Верстка адаптивного інтерфейсу на HTML/CSS;
- Впровадження фільтрації меню й обробки форм на JavaScript.

Об’єктом дослідження є процеси створення вебсайту ресторанного бізнесу: інформування, бронювання, замовлення і взаємодія користувача.

Предметом дослідження є методи, інструменти та технології UX/UI-дизайну й реалізації динамічного контенту.

Методи дослідження аналіз наукової літератури й веб джерел; порівняльний аналіз сайтів конкурентів; прототипування у Figma;

експериментальне тестування HTML/CSS адаптивності; імплементацію JavaScript функцій для фільтрації й форм; оптимізацію продуктивності.

Базою дослідження є прикладні матеріали сайтів-конкурентів (“Пузата Хата”, “Львівські Круасани”, “Сушія”), MDN WebDocs.

Наукова значимість полягає в розробці комплексної методики створення односторінкових вебсайтів для ресторанного бізнесу, що інтегрує теоретичні основи UX/UI-дизайну, адаптивної верстки та динамічного контенту. Запропонована структура дослідження та набір завдань які можуть слугувати шаблоном для подальших наукових робіт у галузі веброзробки та цифрового маркетингу HoReCa (Hotel, Restaurant, Café).

Практична значимість полягає в тому, що результати роботи – готовий прототип, технічне рішення та рекомендації з оптимізації – можуть бути безпосередньо впроваджені в діяльність ресторанів, що прагнуть підвищити рівень онлайн-залучення клієнтів, скоротити час бронювання і збільшити обсяги замовлень. Крім того, напрацювання можуть бути використані ІТ-компаніями та фрілансерами як ефективний шаблон при розробці подібних односторінкових лендінгів.

Кваліфікаційна робота містить 46 сторінок, 39 рисунків та 43 найменування використаних джерел.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ СТВОРЕННЯ ВЕБСАЙТІВ

1.1. Історія вебтехнологій

Мемекс є першим кроком до сучасного вебу. Ця розробка взяла свій початок ще у 1945 році американським інженером та науковцем Ваннівара Буша та стала потужним теоретичним фундаментом для майбутнього розвитку інтернет-технологій. В основі цієї ідеї лежить нелінійне зберігання інформації та доступ до неї. Цей пристрій Ваннівара вирішив назвати Мемекс, (скорочення від Memory Extender – “Розширювач пам’яті”).

Мемех, як його уявляв Ваннівара Буш у своїй новаторській статті 1945 року “Як ми можемо думати”, – що захоплює поєднання уяви та передбачення. Її метою було сприяти зберігання та пошуку інформації в організований та ефективний спосіб [43]. Хоча Мемекс був теоретичним пристроєм і ця концепція так і не була втілена на практиці за браком технічних можливостей того часу, але це стало поштовхом для розвитку інтернету та сучасних технологій. Мемекс можна було порівняти з механізованою бібліотекою, яка дозволяла її користувачам зберігати й сортувати інформацію, для того, щоб при потребі дуже швидко та зручно мати доступ до неї.

Мемекс був спроектований як робочий стіл, в який було вбудовано пристрій для запису та читання мікрофільмів, яким керував набір електромеханічних засобів. При потребі користувач міг додавати та видаляти записи з мікрофільмами. Тобто завдяки Мемексу з’явилася можливість встановлювати логічні зв’язки між різними частинами мікрофільмів та швидкого керування ними. Саме ця інновація того часу була подібною до гіперпосилань, які ми маємо та використовуємо сьогодні.

Мемех забезпечує зберігання та систематизацію інформації таким чином, щоб імітувати асоціативну пам’ять людини. Це означає, що він може знаходити та відображати пов’язаний вміст на основі асоціацій, які користувач має між різними фрагментами інформації.

Мемекс – це теоретичний прийом, спрямований на покращення людської пам’яті та знань, дозволяючи людям зберігати та згадувати інформацію механічним способом [33].

Наступним кроком до WWW є гіпертекст та гіперпосилання. В 1965 році Теодор Нельсон вперше використав у своїх наукових статтях термін “гіпертекст”, цією назвою він описав нелінійний спосіб організації інформації, де користувач міг за допомогою гіперпосилань переміщатися між взаємопов’язаними фрагментами тексту, що допомагало мати більш гнучку та швидку навігацію та переходити на інші пов’язані ресурси. На вебсторінці гіперпосилання зазвичай виглядають як текст або зображення, на які можна натиснути, та часто підсвічуються синім кольором. Саме гіперпосилання в сучасному інтернеті є базовим елементом HTML–документа. Гіперпосилання у HTML–документі може вказувати як на файл, що лежить у тій самій директорії на сервері, так і вміщати повний шлях URL до файлу, який розташований в Інтернеті. Також можна створювати гіперпосилання в середині одного документа, на електронну пошту, відео, аудіофайли та інші документи [31].

У 1980 році Тім Бернерс-Лі, який працював в CERN, запровадив систему ENQUIRE, саме ця система допомогла вченим швидко та зручно обмінюватися між собою документами. Трошки пізніше, у 1989 році Тім Бернерс-Лі запропонував ідею Всесвітньої павутини (WWW). Ця система містить в собі взаємопов’язані між собою документи та різні ресурси, які можна знайти в Інтернет. Вона працює завдяки гіпертексту та гіперпосиланням, що дозволяють переходити від однієї сторінки до іншої.

Для коректної роботи цієї системи (WWW) Тім Бернерс-Лі створив три важливі речі: URI – адреси для вебсторінки, HTTP – спосіб передавання даних, HTML – мову для створення вебсторінки. Тим самим, створення гіпертексту та гіперпосилань стало найважливішим винаходом для розвитку Всесвітньої павутини, що зробило її ефективним засобом для передачі та отримання інформації.

Сайт для внутрішнього користування info.cern.ch з'явився 20 грудня 1990 року. А першою у світі адресою вебсторінки була <http://info.cern.ch/hypertext/WWW/TheProject.html> [30].

1.2. Виникнення комп'ютерних мереж і ARPANET – перший у світі вебсайт

Виникнення комп'ютерних мереж тісно пов'язане з розвитком мережі ARPANET, з якої, як вважають, почався розвиток Інтернету. Ціль її створення полягала в тому, щоб розробити надійну комунікаційну систему. Ця мережа була розроблена військовими, науковими інститутами та Міністерством оборони США. Найголовніша мета полягала в тому, щоб мати стабільний та надійний зв'язок між ними, щоб мати можливість постійно обмінюватись інформацією, а також щоб не втратити ніякі важливі дані навіть після атомного удару, бо тривала Холодна війна.

Важлива особливість мережі ARPANET була в тому, що вона передавала дані не безперервним потоком, а розділяла їх на невеликі пакети, адже вони були більш керовані. Саме це допомогло забезпечити більшу надійність та масштабованість мереж, тому що інформація могла передаватись не єдиним каналом зв'язку, а різними маршрутами.

ARPANET покладалася на два основні протоколи для своєї роботи: Transmission Control Protocol (TCP) та Internet Protocol (IP). TCP забезпечував надійне доставлення даних шляхом ділення їх на пакети, передачі та збирання в правильному порядку на приймальному кінці. IP, з іншого боку, керував адресацією та маршрутизацією пакетів, гарантуючи їх досягнення призначеної мети [2].

Наступним кроком стало створення в 1971 році першого програмного забезпечення, за допомогою якого можна було обмінюватись даними (CPYNET) між комп'ютерами користувачів, підключеними до ARPANET. Цю програму розробив Роберт Томлінсон. Вже невдовзі до цієї програми він додав нову функцію – обміну текстовими повідомленнями між комп'ютерами. Повідомлення можна було надсилати вказуючи адресу отримувача, через

командний рядок, для цього потрібно було використати символ “@”. Саме ця розробка була першою версією системи електронної пошти. В той час це стало значним кроком вперед, адже це сильно полегшило комунікацію між науковцями, дослідниками, інженерами, а згодом було також створено дискусійні форуми, поштові розсилки та інше.

Ранні мови розмітки SGML та HTML заклали основу формування сучасної мережі Інтернет. Але передавати інформацію – це тільки перший крок, важливо також її структурувати. Саме тому були розроблені мови розмітки, вони є фундаментом для створення вебсторінок, а саме у 1986 році SGML (Standard Generalized Markup Language) і HTML у 1991 році [35]. (HyperText Markup Language). Ці мови розміток стали стандартами для розмітки документів. Вони дозволили структурно організувати інформацію на вебсторінках та забезпечити можливість інтерактивного доступу до різних видів контенту.

SGML чітко визначає загальний синтаксис введення елементів розмітки до документа, а також окремі синтаксичні правила вживання тегів (DTD). Це дозволяє авторам створювати та використовувати різноманітну розмітку, обираючи найвідповідніші за змістом теги, та називаючи їх зрозумілою своєю мовою. Таким чином, SGML вірніше буде назвати метамовою, з якої походять багато окремих мов розмітки. З кінця 80-х років більшість нових важливих мов розмітки, таких як TEI та DocBook, базуються на системі мови SGML. У 1986 році SGML була проголошена ISO як міжнародний стандарт мови розмітки (ISO 8879) [34].

Не дивлячись, що ця метамова виявилась дійсно складною для вивчення та застосування на практиці, вона виявилась гнучкою. Саме тому вона використовується в галузях, де потрібно опрацьовувати великі обсяги документів, зокрема в науці, та правознавстві. У SGML було необов'язково закривати теги, або навіть пропускати два. Це дозволило користувачам економити час та сили, та спростити процес розмітки, особливо, у великих текстових масивах. Але для того, щоб використовувати SGML, потрібно було мати спеціальні знання, адже розробники повинні були створювати свої власні

схеми (DTD – Document Type Definition), встановлювати правила для тегів і атрибутів, що було не так просто.

Ситуація змінилася, коли Сер Тім Бернерс-Лі дізнався про SGML від свого колеги Андерса Берглунда та інших з CERN та використав синтаксис SGML для створення HTML. HTML схожа на інші “тегові” мови, засновані на SGML, хоча вона почала свій розвиток, як найпростіша з усіх, до того ж з незакінченим офіційним Визначенням Типу Документа (DTD) [18].

Відмінність від SGML полягала в тому, що використовуючи HTML не можна створювати власні складні схеми для документа, а потрібно використовувати лише невеликий набір стандартних тегів, наприклад як `<h3>`, `<p>`. Звичайний користувач не може їх бачити, адже ці теги використовуються для інтерпретації вмісту сторінки. Через те, що HTML був спрямований на розробку вебсторінок, він надає можливість створювати таблиці, вставляти гіперпосилання, додавати текст та зображення. Крім того, можна створювати структурні документи, використовуючи теги HTML, завдяки яким можна визначити різні частини сторінки, наприклад, заголовки, абзаци, списки. Використання спеціальних тегів, робить зміст ще більш зрозумілим як користувачам, так і комп’ютерним системам.

1.3. JavaScript та його фреймворки: домінування у веброзробці

З розвитком вебтехнологій зросла і потреба в інтерактивності та функціональності вебдодатків. Саме JavaScript (JS) є однією з найважливіших мов програмування в сучасній веброзробці.

JavaScript – це інтерпретована високорівнева мова програмування, яка підтримує імперативний, функціональний і подієво-орієнтовані стилі. Мова програмування з динамічною типізацією. На синтаксис мови програмування JavaScript вплинули мови C/ C++ і Java. Мова програмування, офіційно представлена була ще 1995 року. Свої позиції стрімко JS зайняла у зв’язку з розвитком AJAX технології. Коли зробила можливим оновлення інформації на сторінці без перезавантаження браузера [40].

Саме ця мова дає можливість взаємодії з елементами вебсторінки, можна змінювати їх вміст, стилі та поведінку, створювати різні складні веб-додатки, анімації та інтерактивні карти. Наприклад, на JavaScript можна створити сценарій з умовою, що через 3 секунди перебування користувача на сайті йому показується рекламне сповіщення в модальному вікні. Або за натисканням на кнопку “показати ще” завантажується додаткова група товарів [42].

Крім того, JavaScript також став фундаментом для розвитку додатків, які складаються лише з однієї сторінки (SPA), де вся взаємодія з користувачем відбувається в реальному часі, без перезавантаження сторінки.

Але популярність JS також підкріплюється фреймворками та бібліотекою, наприклад як: React, Angular, Vue.js. Саме вони роблять процес розробки спрощеним та прискореним, а також дозволяють писати код для сучасних додатків більш структурованим. Вибір фреймворк залежить від технічних вимог проєкту, досвіду команди, кожен із них має як переваги та недоліки.

РОЗДІЛ 2. ВЕБСАЙТИ РЕСТОРАННОГО БІЗНЕСУ ТА ЇХ ОСОБЛИВОСТІ

2.1. Ціль ресторанних сайтів та їх типи

Швидкий темп життя спонукає нас частіше харчуватися в ресторані, або кафе, та й додому ми все частіше замовляємо готові страви. Тому дуже важливо, щоб сайти для ресторанів задовольняли потреби користувачів. Розвиток технологій також мав вплив і на вебсайти ресторанного бізнесу та на їх взаємодії з потенційними клієнтами. Якщо раніше ресторанні сайти містили лише найголовнішу інформацію, то з часом вони перетворилися на динамічні, інтерактивні платформи. Завдяки яким ресторан може ефективно взаємодіяти з клієнтами, залучати нових відвідувачів та підвищувати прибутковість. Тим самим вебсайти ресторанного бізнесу є не тільки візиткою закладу, але й потужним інструментом для пошуку клієнтів. Існує багато різних видів вебсайтів для взаємодії з клієнтами, вибір якого залежить від специфіки ресторану та його цілі. Наприклад якщо ресторан націлений на місцевих клієнтів, то може бути достатньо звичайного інформаційного сайту. Він містить тільки найголовнішу інформацію: меню, години роботи, адреса та контактні дані. Зазвичай він не має ніяких додаткових послуг.

Однак, якщо ресторани хочуть збільшити обсяг продаж або доставлення, важливо мати інший тип ресторанного сайту з розширеним функціоналом, який охоплює підтвердження та відмову бронювання. Одним із найпопулярніших типів сайту – це сайти для онлайн-замовлення їжі, адже вони дозволяють клієнтам мати швидкий доступ до меню. Клієнт може самостійно подивись склад страв, оформити замовлення на дім, або забрати самостійно. Також в останній час стали популярними сайти з інтеграцією чат-ботів. Адже в їх основі лежать автоматизовані системи, які дозволяють ресторану обробляти запити клієнтів онлайн. Вони обробляють запити на бронювання, замовлення або запитання щодо меню. Їх сильна сторона в тому, що вони значно полегшують взаємодію з клієнтами, оскільки вони можуть надати миттєві відповіді на питання.

Але крім того, успіх проєкту залежить від якості Web-ресурсу. Деякі власники закладів займаються розробкою сайтів самостійно, орієнтуючись на конструктори [23], розміщені в інтернеті. Виконати роботу без певних знань складно, і найчастіше такі представництва у мережі успіху немає. Незручне меню, слабка швидкість, фотографії низької якості відлякують користувачів, вони залишають сторінку та переходять до конкурентів.

UX/UI дизайн визначає, наскільки зручною, інтуїтивною та естетично привабливою буде взаємодія користувача з ресторанним сайтом. Крім технічної сторони сайту, також велику роль грає якісний дизайн. Адже саме зручність та привабливість дизайну вирішує загальне враження гостя від сайту, що безпосередньо буде мати вплив і на сам ресторан. А це означає, що вебсайт повинен бути максимально зрозумілим, з простою навігацією. Адже якщо відвідувач не зможе зрозуміти як зробити онлайн бронювання, адресу, або меню, то це може призвести до втрати клієнта. Саме тому основна інформація повинна бути в швидкому доступі. В цьому може допомогти UX/UI дизайн. Що це?

UX-дизайн (User Experience) – в основі цього дизайну лежить забезпечення легкого, інтуїтивного та зручного користування вебсайтом. Це залежить від того, який шрифт був обрано, кольори, іконки. Також важливим аспектом UX-дизайну є розуміння яку мету мають користувачі вебсайту. Саме це допоможе створити такий дизайн, який допоможе збільшити кількість гостей. Наприклад, якщо найголовніша мета гостей – це онлайн-замовлення їжі, то саме цей розділ має знаходитись на самому видному місці, та бути легким у використанні.

Деякі основні принципи UX-дизайну:

1. Доступність. Користувач без потреби в додаткових поясненнях розуміє як використовувати вебсайт.
2. Погодженість. Вона виражається в єдиному стилі та дизайні всіх елементів інтерфейсу. До цього відноситься використання однакових шрифтів, кольорів, елементів управління.
3. Орієнтованість на користувача. Розробляючи UX/UI важливо пам'ятати про очікування цільової аудиторії продукту та її переваги. Зрозуміти їхні вподобання допомагає аналітика – від опитувань до теплових карт.

4. Впізнаваність бренду. Добре продуманий дизайн може допомогти створити впізнаваний бренд і виділити його на ринку [36].

Адаптивність та мобільна оптимізація є ключовими факторами успіху ресторанного сайту, оскільки більшість користувачів заходять зі смартфонів і очікують коректного відображення й швидкого завантаження на будь-якому пристрої. У сучасних умовах розвитку цифрових технологій статистика свідчить, що більшість користувачів відвідують сайти зі смартфонів, ніж з планшетів, або комп'ютерів, з цього випливає необхідність створення зручного інтерфейсу для всіх типів пристроїв. Головна мета власника – це продати послугу чи товар найвигіднішим способом. Тільки якщо користувач не матиме можливості побачити правильне відображення вебсайту, малоймовірно, що він зробить те, що планував. А це може призвести до втрати потенційних клієнтів, погіршення рейтингу. Саме тому адаптивність та мобільна оптимізація вебсайтів є дуже важливою частиною правильного функціонування вебсайтів (рис. 2.1).



Рис. 2.1 Адаптивний дизайн [25]

2.2. Аналіз наявних рішень

Для аналізу відібрано вебсайти закладів, знайомі кожному українцю. Спочатку починаємо з вебсайту мережі ресторанів “Пузата Хата”.

Заходячи на головну сторінку, користувач бачить модальне вікно, де він повинен обрати місто та найближчий до нього заклад мережі. Саме це допомагає бачити тільки ті страви, які є в наявності, саме в обраному закладі. Вебсайт є

дуже простим та зрозумілим в користуванні, адже має чітко організовану структуру та тільки необхідні підкатегорії. Крім того, вебсайт дає можливість вибрати потрібну мову, що може допомогти в збільшенні кількості клієнтів. Також бічне меню містить такі важливі категорії як: “Змінити заклад”, “Контакти”, “Ми в соцмережах”, “Корисна інформація”. Останній пункт є дуже важливим, тому що в ньому є підкатегорія, яка перераховує продукти, які можуть викликати алергічні реакції. Користувач може ознайомитись зі складом страви та бути уважним при її виборі. Також опис до кожної страви містить в собі таку інформацію як: Енергетична цінність на 100 г, білки, жири та вуглеводи. Це розраховано на категорію клієнтів, які орієнтовані на здорове та повноцінне харчування (рис.2.2).

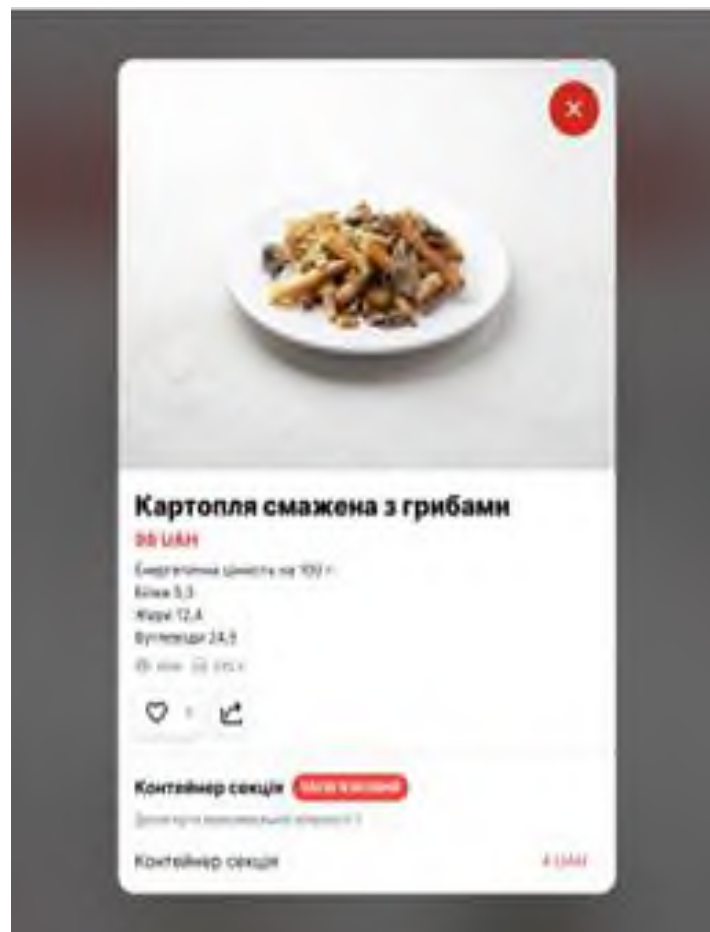


Рис. 2.2 Опис страви [37]

Також вебсайт оптимізовано для мобільних пристроїв, що дозволяє користувачам замовляти їжу додому, на самовивіз, або просто переглядати меню

прямо зі смартфонів. Така зручність користування вебсайтом орієнтована на збільшення кількості клієнтів і це дає результат.

Наступний заклад, на вебсайт якого звернемо увагу це вебсайт мережі кафе “Львівські Круасани”. Хоча вебсайт містить в собі набагато більше категорій та підкатегорій меню, але сучасний UX/UI дизайн, який створений за допомогою привабливих зображень, зрозумілих шрифтів та кнопок, робить його дуже зручним та легким у використанні.

Як і у випадку з “Пузатою Хатою”, вебсайт “Львівських Круасанів” добре оптимізовано для мобільних пристроїв, що дає змогу клієнтам здійснювати покупки на ходу. Також в категорії “Пекарні” є можливість вибрати місто та побачити на карті найближчі заклади, не витрачаючи час на пошук за адресою (рис. 2.3).

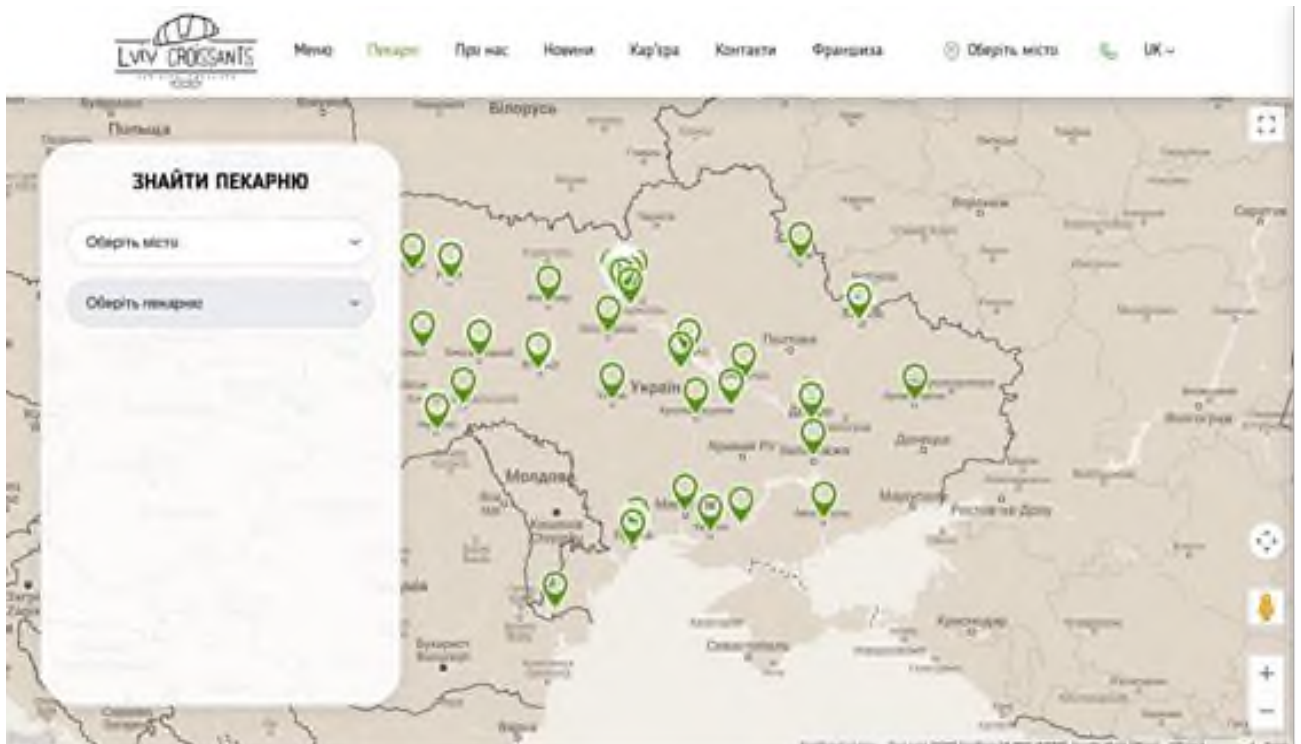


Рис. 2.3 Мапа закладів [21]

Цей сайт є хорошим прикладом того, як важливо залучати й емоційну частину, саме тому в категорії “Про нас” описана історія створення компанії, її здобутки та розвиток.

Наступний вебсайт, який також є ще одним прикладом ефективного використання цифрових технологій у ресторанному бізнесі – це вебсайт

ресторану “Сушія”. На головній сторінці можна відразу побачити актуальну інформацію про акції та знижки (рис. 2.4). Сайт також добре працює на всіх пристроях, що є важливим компонентом зручності користувачів.

Якщо немає можливості завітати в заклад, то клієнти можуть зробити замовлення онлайн або на самовивіз. Для підтримки та заохочення постійних клієнтів, вебсайт включає інтегровану систему бонусної програми. Ця функція сприяє не тільки збільшенню продажів, а й формуванню довготривалих відносин з клієнтами.



Рис. 2.4 Банер зі знижками [38]

В той же самий час може здатися, що вебсайт містить надмірну кількість інформації та інтерактивних елементів, а це все може перевантажувати інтерфейс, ускладнюючи навігацію та використання.

Після огляду конкурентних вебсайтів виконано оцінювання їхніх сильних і слабких сторін.

Для мережі “Пузата Хата” варто відзначити такі аспекти:

- Сильні сторони: Зручне меню, список алергенів, можливість бронювання столиків та доставлення.
- Слабкі сторони: Обмежена програма лояльності.

Для “Львівських Круасанів” характерні наступні особливості:

- Сильні сторони: Програма лояльності, онлайн-замовлення, емоційна частина.

- Слабкі сторони: відсутність особистого кабінету користувача, що робить неможливим побачити історію замовлень.

І для “Сушії” можна виділити такі переваги та недоліки:

- Сильні сторони: Зручність онлайн-замовлення, програма лояльності.
- Слабкі сторони: Сайт занадто насичений функціями, що інколи ускладнює пошук потрібної інформації.

Попри різницю у підходах до дизайну та функціональності, всі ці сайти мають спільні риси:

1. Адаптивність та мобільна оптимізація. Всі сайти добре відображаються на мобільних пристроях, що є важливим у сучасному світі, де більшість користувачів здійснюють пошук та покупки через смартфони.

2. Інтерфейс для користувачів. Простота навігації, доступ до меню, можливість швидкого оформлення замовлення – це важливі фактори для зручності клієнтів.

3. Система самовивозу. Майже всі ресторани мають можливість замовлення їжі на самовивіз.

Таким чином, конкурентні сайти ресторанів орієнтуються на потреби користувачів і використовують сучасні вебтехнології для покращення клієнтського досвіду.

2.3. Вимоги до вебсайту, технічні та функціональні

Для оптимальної роботи вебсайту було б правильно зосереджуватись на технічних та функціональних вимогах. Що сприятиме коректній та безперебійній роботі вебсайту, адаптивності до різних пристроїв, а також використання інноваційних інструментів. Наприклад: бронювання столиків та замовлення онлайн, чат-боти. А це, своєю чергою, відповідає сучасним стандартам та задовольняє потреби користувачів у сфері ресторанного бізнесу.

Технічні вимоги – це набір специфікацій, стандартів та критеріїв, які визначають базові параметри функціонування вебсайту з погляду технологій,

безпеки та ефективності роботи. Саме ці вимоги є основою, на якій ґрунтується робота всієї системи, забезпечуючи її надійність, масштабованість та оптимальну продуктивність. Наприклад:

Адаптивність та мобільна версія:

- Мобільна оптимізація. Сайт повинен бути адаптованим до різних розмірів та автоматично підлаштовуватися до екранів (мобільних телефонів, планшетів, комп'ютерів).
- Тестування на різних пристроях. Важливо перевірити як виглядає та чи правильно функціонує сайт на основних операційних системах, таких як: iOS, Android.

Швидкість завантаження та оптимізація:

- Швидке завантаження. Дуже важливо, щоб вебсайт завантажувався швидко, не більше 3-х секунд, це може залучити більшу кількість клієнтів та призведе до нижчих показників відмов.
- Оптимізація коду. Починається із проведення ретельної перевірки коду, щоб виявити зайвий або роздутий код. Використовуються такі інструменти, як Google PageSpeed Insights або GTmetrix, щоб отримати уявлення про показники продуктивності вебсайту, такі як час завантаження, розмір сторінки та кількість запитів [13].

Мінімізація:

- Зменшення файлів CSS і JavaScript. Також прискорює час в завантаженні сторінок. Може допомогти видалення зайвих символів, пропусків, а також непотрібних коментарів. Тим самим буде зменшено розмір файлу і обсяг даних, які потрібно обробити.

Сумісність з браузером:

- Кросбраузерність. Сайт повинен бути сумісним та коректно працювати з популярними веббраузерами (Google Chrome, Safari, Edge, Opera, Mozilla Firefox), навіть на застарілих версіях.

Функціональні вимоги – це детальні специфікації, які визначають дії, поведінку та функціональні можливості, які повинна виконувати система, щоб відповідати призначеній меті. У контексті програмного забезпечення та систем

вони описують те, що система має робити: включаючи завдання, які система повинна виконувати, як вона взаємодіятиме з користувачами та як вона реагуватиме на різні вхідні дані чи події. [39].

Варіанти взаємодії з користувачами:

- Зворотний зв'язок: в цьому можуть допомогти різні форми для реєстрації, контактні форми та чат-боти, для можливості відповідей у реальному часі без довгих очікувань.
- Соціальні мережі: Можливість авторизації через соціальні мережі або поділитися контентом в себе на сторінці в соціальній мережі, що також може залучити нових гостей.
- Контент: це можуть бути різні каруселі, меню, анімації, а також відео, зображення, що зробить вебсайт більш інформативним та привабливим.
- Мультимовність: налаштування вебсайту, для користувачів, які володіють різними мовами. Саме це допомагає вийти на міжнародний ринок та залучити більшу аудиторію.

РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА СТВОРЕННЯ ВЕБСАЙТУ

3.1. Прототипування та дизайн у Figma

Перед безпосередньою розробкою було створено детальний макет вебсайту в інструменті Figma (рис. 3.1). Розглянемо детально кожен із фреймів.

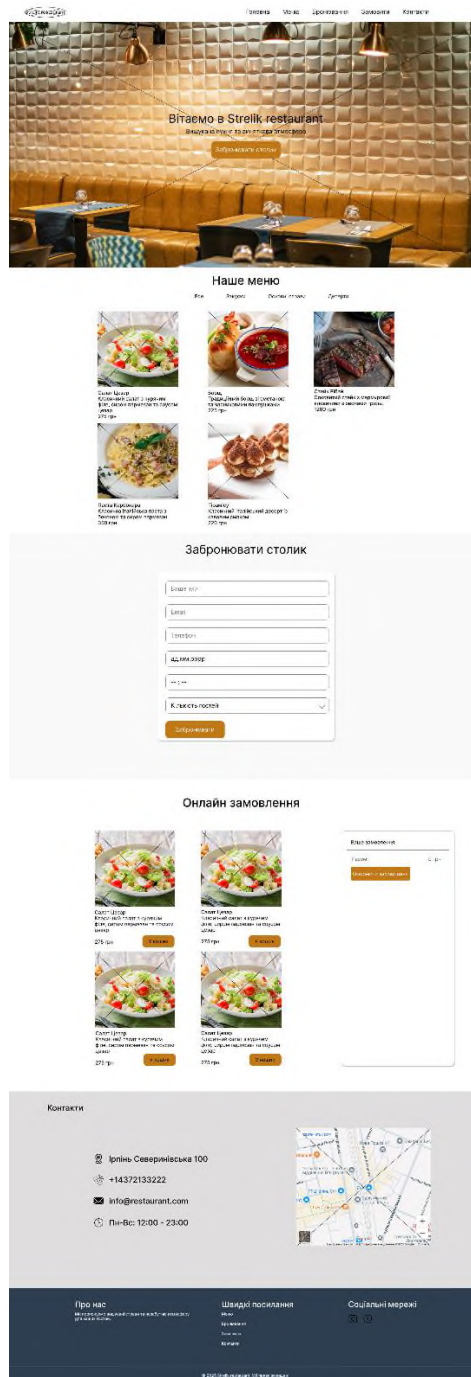


Рис 3.1 Макет вебсайту виконаний у Figma

Frame 1 ілюструє загальний вигляд головної сторінки з великим банером та горизонтальним навігаційним меню, що задає тон усьому інтерфейсу.

Frame 2 демонструє сітку та візуальну ієрархію розділу “Меню”, де опрацьовано розміщення карток страв і механіку фільтрації за категоріями.

Frame 3 містить інтерфейс форми бронювання та онлайн-замовлення зі всіма необхідними полями й підказками.

Frame 4 представляє дизайн кошика та бокової панелі з підсумковою вартістю замовлення, а Frames 5-6 деталізують секцію контактів і футер із вибраною кольоровою палітрою та іконками соціальних мереж.

Під час верстки та реалізації деякі деталі макета були змінені: розташування кнопок навігації уточнено згідно з результатами внутрішнього тестування, а відтінок футера затемнено для покращеної читабельності тексту на всіх пристроях.

Також буде надано покликання на цей файл у Figma: <https://www.figma.com/design/eGXHxAewu4DZfDOsnRMghx/Untitled?node-id=74-54&t=W5XCdkR80oyaSqso-1>.

Мобільну версію сайту реалізовано без окремого прототипу в Figma – для цього в CSS застосовано media queries [26] на точках перелому max-width: 768px і max-width: 480px, відносні одиниці (% , vw, em) для адаптивних відступів і розмірів шрифтів, а основне меню на малих екранах сховано за іконкою “гамбургер” з плавним відкриттям через transform і transition [8]. Також оптимізовано зображення через max-width: 100% та height: auto для коректного масштабування на мобільних пристроях [27].

3.2. Структура сайту

У цьому розділі викладено призначення ключових елементів структури вебсайту. Основною метою створення вебсайту ресторану “Strelik” було не лише інформування відвідувачів про асортимент страв, бронювання столиків, онлайн замовлень і режим роботи, а й створення емоційного враження – бажання завітати до закладу вже з першого перегляду сторінки. Сайт позиціюється як мінімалістичний онлайн-простір, який одразу показує атмосферу ресторану та сучасність.

Спочатку звертаємо увагу на структуру вебсайту та його розділи. Для побудови структури намагалось врахувати побажання потенціальних гостей вебсайту. Саме тому сайт побудований за принципом односторінкового лендінгу (one-page site), де користувач прокручує контент вниз, поступово знайомлячись з усією інформацією.

Але чому такий вибір? Основні переваги лендінг пейджу перед іншими видами сайтів у тому, що вся структура та контент орієнтовані тільки на цільову дію. Відвідувач має 6-10 секунд для визначення цінності сайту, на який він потрапив, щоб вирішити для себе звернутися в цю компанію або продовжувати пошуки. Все коротко, по суті, і точно в ціль [32].

Перший елемент, який може побачити користувач – це хедер (рис. 3.2), або верхнє меню навігації. При розробці мета була створити його простим, але максимально функціональним. Зліва він містить логотип закладу. А справа містить основні пункти меню:

- Головна – посилання, яке повертає на початок сторінки.
- Меню – список доступних страв та їх опис.
- Бронювання – форма для замовлення столика.
- Замовити – можливість онлайн-замовлення страв.
- Контакти – адреса, телефон, карта та посилання на соціальні мережі.

Це допомагає гостю не витратити час на скролінг (вертикальне або горизонтальне переміщення зображення у вікні екрану), а одразу перейти до потрібної категорії.

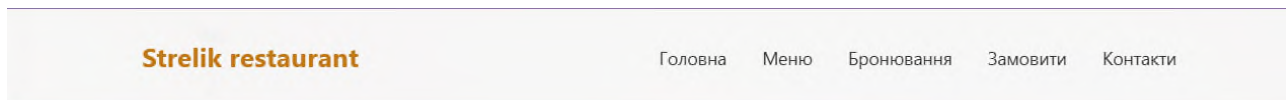


Рис. 3.2 Header

Про те, найперше, що впадає в очі, це банер, на котрому користувач бачить привітальне повідомлення “Вітаємо в Strelik restaurant” (рис. 3.3). Тобто якщо гість має бажання тільки забронювати столик, то він може дуже швидко це зробити. Потрібно лише натиснути на кнопку “Забронювати столик”, і ця дія одразу переведе його до форми бронювання.

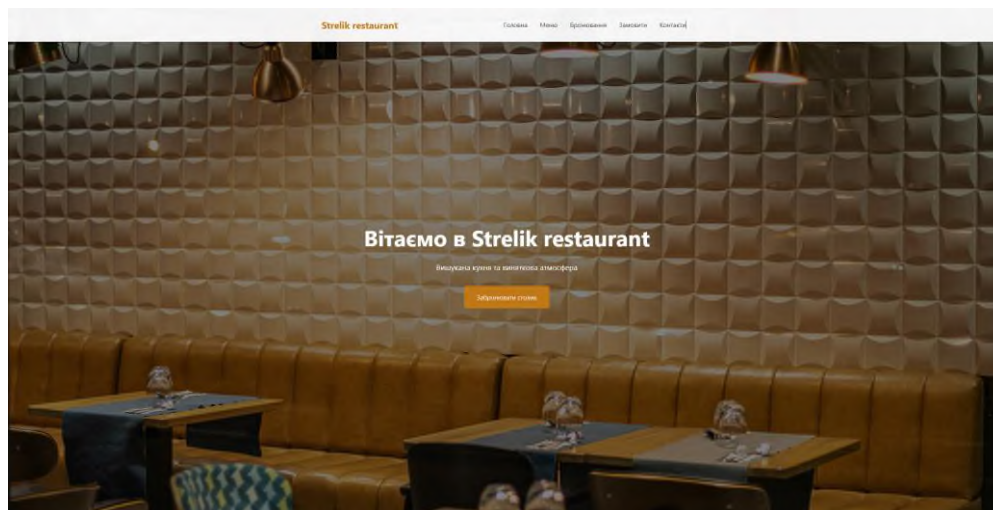


Рис. 3.3 Банер із привітанням

Наступний та найважливіший розділ – це Меню. Воно містить такі категорії як: Всі, Закуси, Основні страви, Десерти. Це один із головних блоків сайту, який розроблений з використанням сучасних підходів до UX/UI та оновлення контенту відбувається динамічно, тобто без оновлення сторінки. Меню розроблено у вигляді табів (tabs, вкладки), за допомогою яких користувач може швидко переміщатися між категоріями страв.

Всі таби мають однаковий вигляд, але активний підсвічується жовтим кольором (рис. 3.4).

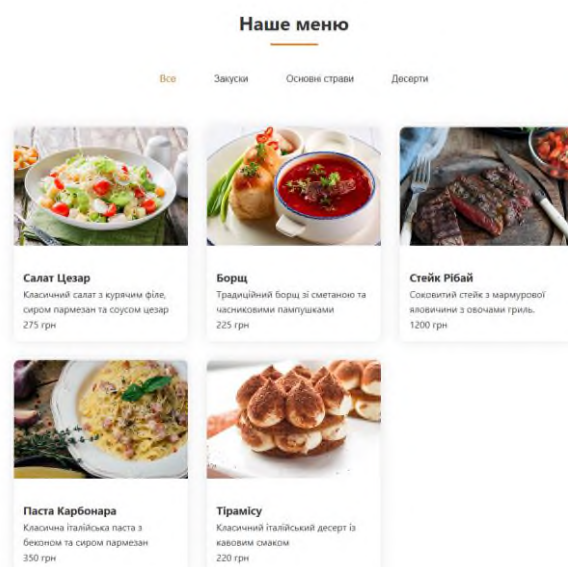


Рис. 3.4 Меню зі стравами

Кожна позиція меню відображається у вигляді картки, яка містить наступні

елементи: зображення, назву, склад та ціну страви (рис. 3.5).



Рис. 3.5 Опис страви

Наступник блок містить в собі форму для бронювання столиків. Ця форма дозволяє швидко та звучно забронювати столик без необхідності телефонного дзвінка, тим самим полегшує обробку заявок адміністрацією закладу. Ця форма бронювання містить в собі такі поля, заповнюючи які, гість надає всю необхідну інформацію для оформлення бронювання. Всі вони є обов'язковими для заповнення:

- Ім'я – текстове поле для введення імені відвідувача.
- Email – поле з типом email має автоматичну перевірку на формат адреси.
- Телефон – поле для номера телефону.
- Дата і час – ці поля дозволяють вибрати бажану дату і час використовуючи календар (рис. 3.6).

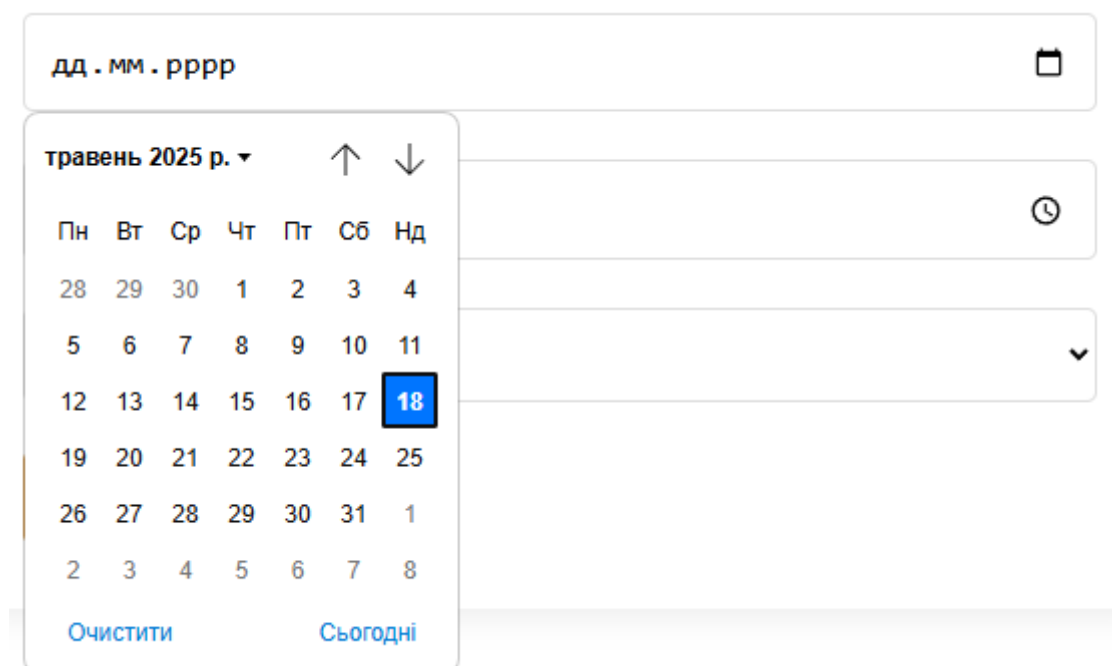


Рис. 3.6 Календар

Кількість гостей – список що в розкривається (select), який містить в собі варіанти (option) вибору кількості гостей (рис. 3.7).

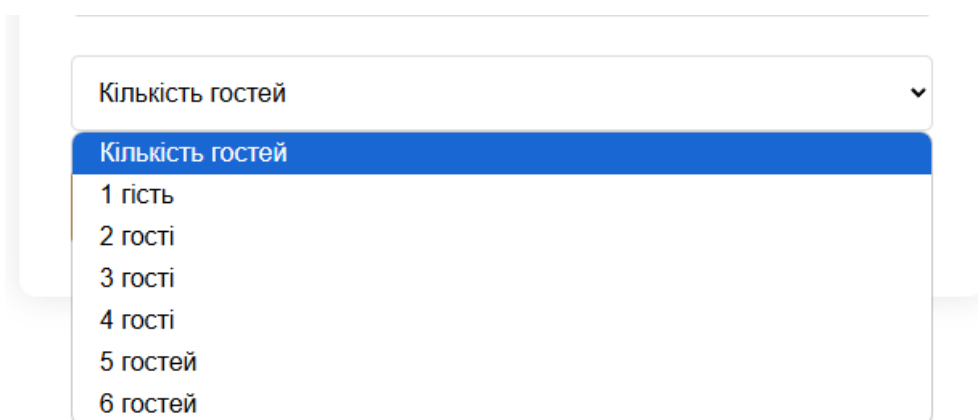


Рис. 3.7 Вибір кількості гостей

Якщо гість правильно заповнив всі необхідні поля, то після натиснення кнопки “Забронювати” він побачить таке сповіщення: (рис. 3.8).

Рис. 3.8 Успішне бронювання столика

Вебсайт дає можливість зробити онлайн-замовлення. Цей розділ дуже подібний до розділу “Меню”, але він дає можливість користувачам не лише переглянути та обрати страву, а й додати обрану позицію до кошика та оформити доставлення (рис. 3.9).

Онлайн замовлення

Ваше замовлення	
Салат Цезар	2 550 грн
Разом до сплати:	550 грн
Час доставки: ~60 хвилин	
Доставка по місту	

Рис. 3.9 Меню для замовлення онлайн

Також справа розташований Блок “Ваше замовлення”, який містить список обраних страв з їх вартістю, а також можливість змінювати кількість страв. Крім того, калькулятор підраховує загальну вартість замовлення в реальному часі (рис. 3.10). Після натискання кнопки “Оформити замовлення” з’являється форма оформлення замовлення (рис. 3.11).

Ваше замовлення

Борщ	- 1 +	225 грн
Салат Цезар	- 1 +	275 грн
Разом до сплати:		500 грн

Час доставки: ~60 хвилин
Доставка по місту

Оформити замовлення

Рис. 3.10 Кошик замовлення

Оформлення замовлення

Контактні дані

Ваше ім'я

Адреса доставки

Квартира/офіс (необов'язково)

Під'їзд (необов'язково)

Поверх (необов'язково)

Телефон

Дані карти

Номер карти

MM/PP CVV

Приблизний час доставки: 1 година
Безпечна оплата

Підтвердити замовлення

Рис. 3.11 Вигляд інтерфейсу форми замовлення

Після успішного оформлення замовлення гість може побачити таке сповіщення: (рис. 3.12).

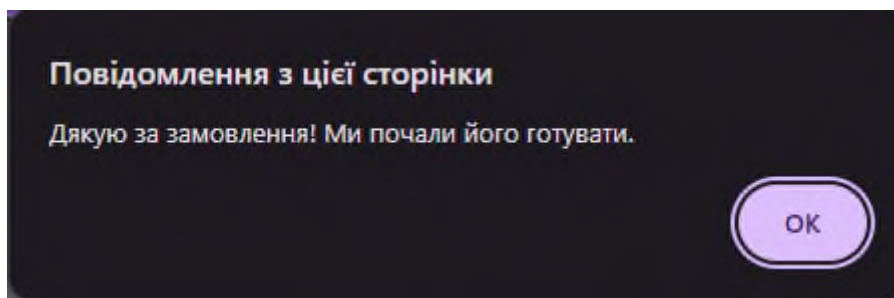


Рис. 3.12 Успішне замовлення страв

Вебсайт також має блок “Контакти”, який розташований майже в кінці головної сторінки та допомагає швидко знайти місце розташування, номер телефону та електронну адресу закладу та години роботи закладу. Також контактні дані розроблені для максимальної зручності гостя. Наприклад: номер телефону зроблений як посилання з префіксом tel:, при натисканні з мобільного пристрою можна одразу зробити виклик. Електронна пошта за допомогою mailto: відкриває поштового клієнта для зручності відправлення листа. А адреса ресторану містить в собі посилання на геолокацію ресторану, тому відразу відкривається Google Maps (рис. 3.13).

Контакти

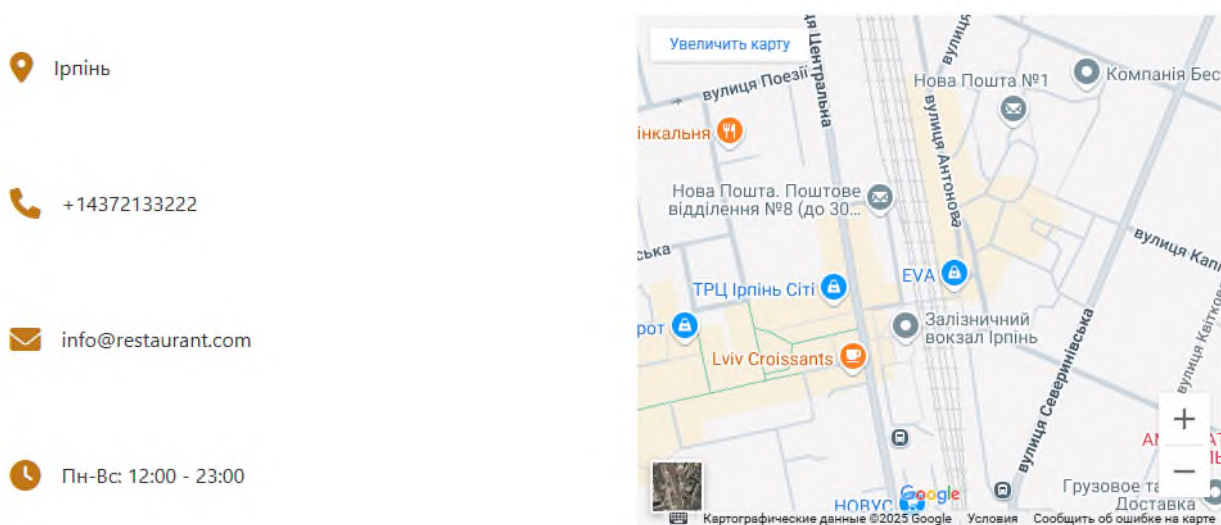


Рис. 3.13 Контактні дані

Кінцевий елемент сайту – це футер. Він надає повторювану інформацію, що допомагає не повертатися до хедера, а швидко перейти до основних розділів: Головна, Меню, Про нас, Бронювання, Контакти, це можливо за допомогою внутрішніх якірних посилань (рис. 3.14).

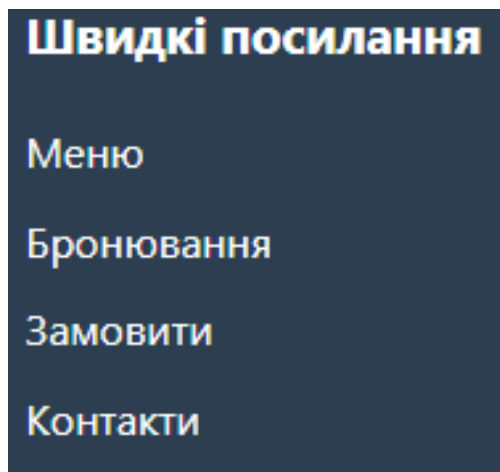


Рис. 3.14 Швидкі посилання

Також футер надає коротку інформацію про заклад і посилання на соціальні мережі (Instagram та Facebook) (рис. 3.15).

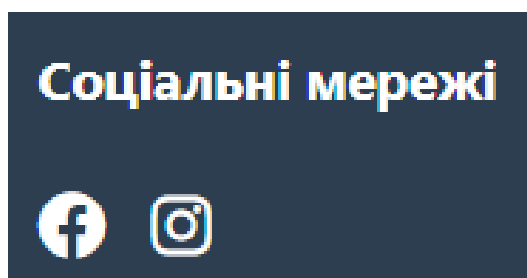


Рис. 3.15 Посилання на соціальні мережі

Забезпечення адаптивності для різних пристроїв передбачає використання медіазапитів, відносних одиниць вимірювання та мобільного меню для коректного відображення сайту на будь-яких екранах. Під час розробки вебсайту приділено увагу адаптивності, завдяки чому відвідувачі, незалежно від пристрою, який вони використовують, можуть бачити коректне відображення вебсайту та користуватися всіма його функціями.

- Медіазапити (media queries) [22] – завдяки їм вебсайт має змогу підлаштовуватись під різні розміри екранів. На вебсайті відбуваються такі зміни, як: розмір шрифтів, відступи та розташування елементів. Наприклад, заголовки на мобільному екрані мають менший розмір.

- Відносні одиниці вимірювання – було вирішено використовувати саме % та vh, замість фіксованих пікселів. Для автоматичного підлаштовування елементів під розмір екрана. А також для зображень (max-width: 100% та height: auto)

- Мобільне меню (“гамбургер”) – на маленькому розмірі екрана основна навігація приховується за іконкою гамбургера. Тільки при натисканні меню на іконку відкривається модальне вікно на весь розмір екрана. Що дозволяє без дискомфорту використовувати вебсайт з мобільного пристрою (рис. 3.16-3.17).



Рис. 3.16 Мобільне меню



Рис. 3.17 Відкрита мобільна версія

3.3. Вибір технологій для розробки

У цьому розділі проведено аналіз сучасних технологій, що застосовуються у веброботці.

Під час розробки вебсайту “Strelik” було проаналізовано сучасний стек технологій. Та обрані базові інструменти такі, як: HTML, CSS та JavaScript. За допомогою них створювалась структура вебсайту, додані стилі та динамічна поведінка сторінок.

HTML (HyperText Markup Language) – це стандартизована мова розмітки документів для перегляду вебсторінок у браузері. Елементи HTML є

будівельними блоками сторінок HTML. За допомогою конструкцій HTML, зображення та інші об'єкти, такі як інтерактивні форми, можуть бути вбудовані у візуалізовану сторінку. HTML надає засоби для створення структурованих документів, позначаючи структурну семантику тексту, наприклад заголовки, абзаци, списки, посилання, цитати та інші елементи. Для побудови структури проєкту використовувалися такі теги: `<header>`, `<section>`, `<footer>`, `<nav>`. Крім того, також були використані елементи: кнопки `<button>`, посилання `<a>`, різні форми, як: `<form>`, `<input>`, `<textarea>`, а також зображення `<img>`.

Наступним і не менш важливим інструментом став CSS [41]. CSS (Cascading Style Sheets) – це спеціальна мова (мова стилів), за допомогою якої описують вигляду документів (як і де відображати елементи вебсторінки), написаних мовами розмітки даних. Якщо сказати просто, то це мова стилів, саме використовуючи CSS можна змінити вигляд сторінки: колір, шрифт, ширина та висота блоків, або відступів.

JavaScript – це мова програмування, яка працює у веббраузері користувача. Коли вебсторінка завантажується, вбудований інтерпретатор JavaScript отримує та виконує JavaScript-інструкції, записані на сторінці. Інтерпретатор читає та виконує JavaScript-інструкції по черзі, перетворюючи їх на машинний код. Це дозволяє JavaScript взаємодіяти зі структурою та вмістом вебсторінки, змінювати HTML та CSS [41], обробляти події, виконувати асинхронні запити на сервер та оновлювати сторінку без перезавантаження.

3.4. Функціональність вебсайту

Розглянемо ключові інтерактивні елементи сайту: кнопки, форми, анімації та ефекти, що забезпечують динамічну взаємодію користувача з ресурсом. Звертаємо увагу на інтерактивні елементи, які були додані під час розробки, адже вони безпосередньо впливають на зручність та взаємодію користувача з вебсайтом.

На вебсайті кнопки `<button>` використовуються, як перехід між категоріями меню, для запуску інших дій (модальних вікон та відправлення форм), а також для відкриття мобільного меню (рис. 3.18-3.26) [1].

```

<div class="menu-categories">
  <button class="category-btn active" data-category="all">Все</button>
  <button class="category-btn" data-category="starters">Закуси</button>
  <button class="category-btn" data-category="main">Основні страви</button>
  <button class="category-btn" data-category="desserts">Десерти</button>
</div>

```

Рис. 3.18 Кнопки для переходу між категоріями меню

Все Закуси Основні страви Десерти

Рис 3.19 Результат переходів між категоріями меню

Далі наведено JavaScript-код, який реалізує цю функціональність фільтрації меню без перезавантаження сторінки (рис. 3.20).

```

// фільтрація меню
categoryButtons.forEach(button => {
  button.addEventListener('click', () => {
    categoryButtons.forEach(btn => btn.classList.remove('active'));
    button.classList.add('active');

    const category = button.dataset.category;
    displayMenuItems(category);
  });
});

function displayMenuItems(category = 'all') {
  const filteredItems = category === 'all'
    ? menuItems
    : menuItems.filter(item => item.category === category);

  menuContainer.innerHTML = filteredItems.map(item => `
    <div class="menu-item">
      
      <div class="menu-item-content">
        <h3>${item.name}</h3>
        <p>${item.description}</p>
        <div class="menu-item-footer">
          <span class="price">${item.price} грн</span>
        </div>
      </div>
    </div>
  `).join('');
}

```

Рис 3.20 JavaScript-код фільтрації та динамічного відображення елементів меню

У цьому фрагменті коду кнопки вибираються за допомогою `document.querySelectorAll` [9], а потім до них через `addEventListener('click')` додається обробник події [11]. При натисканні з усіх кнопок спочатку видаляється клас `active`, а потім додається тільки до натиснутої. Значення категорії зчитується через `button.dataset.category` [19] і передається у функцію `displayMenuItems`, яка фільтрує масив `menuItems`

методом `Array.prototype.filter` [3][4]. Отримані елементи перетворюються у HTML за допомогою `Array.prototype.map` [5] і вставляються в DOM через `element.innerHTML` [10], що забезпечує динамічне оновлення меню без перезавантаження сторінки.

```
<div class="burger">
  <div class="line1"></div>
  <div class="line2"></div>
  <div class="line3"></div>
</div>
```

Рис. 3.21 Кнопка для відкриття мобільного меню



Рис 3.22 Результат кнопки для відкриття мобільного меню

```
<button class="btn checkout-btn">
  <i class="fas fa-shopping-cart"></i>
  Оформити замовлення
</button>
```

Рис 3.23 Кнопка для запуску модального вікна оформлення замовлення



Рис 3.24 Результат кнопки для відкриття модального вікна

```
<div class="cart-item-controls">
  <button class="quantity-btn minus" data-id="${item.id}">-</button>
  <span>${item.quantity}</span>
  <button class="quantity-btn plus" data-id="${item.id}">+</button>
</div>
```

Рис 3.25 Кнопки для зміни кількості товару в кошик



Рис 3.26 Результат кнопок для зміни кількості товарів

Вебсайт має дві форми <form>, які дозволяють гостям ресторану замовити столик, або зробити онлайн замовлення. Також використана HTML валідація через атрибути required [24], maxlength [15], type [17]. Наведена форма бронювання столика на вебсайті “Strelik restaurant” (рис. 3.27).

- required – поле обов’язкове для заповнення.
- maxlength – обмежує максимальну кількість символів.
- type – визначає тип поля (наприклад, text, email, tel, date, time).
- pattern – регулярний вираз для додаткової валідації [16].

```
<form id="reservation-form" class="reservation-form">
  <div class="form-group">
    <input type="text" id="name" name="name" required placeholder="Ваше ім'я">
  </div>
  <div class="form-group">
    <input type="email" id="email" name="email" required placeholder="Email">
  </div>
  <div class="form-group">
    <input
      type="tel" id="phone" name="phone" required placeholder="Телефон"
      pattern="^\+380[ \-]?[0-9]{2}[ \-]?[0-9]{3}[ \-]?[0-9]{4}$"
      title="Формат: +380 XX XXX XXXX"
    >
  </div>
  <div class="form-group">
    <input type="date" id="date" name="date" required>
  </div>
  <div class="form-group">
    <input type="time" id="time" name="time" required>
  </div>
  <div class="form-group">
    <select id="guests" name="guests" required>
      <option value="">Кількість гостей</option>
      <option value="1">1 гість</option>
      <option value="2">2 гості</option>
      <option value="3">3 гості</option>
      <option value="4">4 гості</option>
      <option value="5">5 гостей</option>
      <option value="6">6 гостей</option>
    </select>
  </div>
  <button type="submit" class="btn">Забронювати</button>
</form>
```

Рис. 3.27 Форма бронювання столика

Після HTML-розмітки форми (рис. 3.27) наведено JavaScript-фрагмент (рис. 3.28), який знаходить форму за id=“reservation-form”, додає обробник події на відправлення, збирає дані з полів через new FormData(reservationForm) [12], виводить повідомлення про успішне бронювання через alert() [29] та очищає форму за допомогою reservationForm.reset() [20] без перезавантаження сторінки.

```
// форма бронювання
reservationForm.addEventListener('submit', (e) => {
  e.preventDefault();

  const formData = new FormData(reservationForm);
  const reservation = {
    name: formData.get('name'),
    email: formData.get('email'),
    phone: formData.get('phone'),
    date: formData.get('date'),
    time: formData.get('time'),
    guests: formData.get('guests')
  };

  // повідомлення
  console.log('Деталі бронювання:', reservation);
  alert('Дякую за бронювання! Ми зв'яжемося з вами для підтвердження.');
```

```
reservationForm.reset();
});
```

Рис 3.28 JavaScript-код обробки форми бронювання

Рис 3.29 Результат форми бронювання столика

Після реалізації логіки форми бронювання (рис. 3.27-3.29), де дані збираються через `FormData`, відображається сповіщення та виконується `reset()` без перезавантаження, для онлайн-замовлення (рис. 3.30) застосовується схожий підхід: при події `submit` створюється об'єкт `orderDetails` з полів форми, виводиться повідомлення через `alert()`, закривається модальне вікно, очищується масив `cart`, викликається `updateCart()` для оновлення відображення та виконується `orderForm.reset()`.

```

orderForm.addEventListener('submit', (e) => {
    e.preventDefault();

    const formData = new FormData(orderForm);
    const orderDetails = {
        name: formData.get('name'),
        address: formData.get('address'),
        apartment: formData.get('apartment'),
        entrance: formData.get('entrance'),
        floor: formData.get('floor'),
        phone: formData.get('phone'),
        cardNumber: formData.get('cardNumber'),
        cardExpiry: formData.get('cardExpiry'),
        cardCVV: formData.get('cardCVV'),
        items: cart
    };

    console.log('Деталі замовлення:', orderDetails);
    alert('Дякую за замовлення! Ми почали його готувати.');
```



```

    orderModal.style.display = 'none';
    cart = [];
    updateCart();
    orderForm.reset();
});

```

Рис 3.30 JavaScript-код обробки форми замовлення

На рис. 3.11 представлено інтерфейс онлайн-форми замовлення, що складається з двох основних блоків:

1. Контактні дані – поля для введення імені, адреси доставлення, нюансів (квартира, під'їзд, поверх) та телефону з валідацією формату.
2. Платіжні дані – поля для введення номера картки, терміну дії (MM/YY) та CVV з обмеженням кількості символів і підказками щодо безпечної оплати.

Під формою відображаються індикатори приблизного часу доставлення та безпечності платежу, а кнопка “Підтвердити замовлення” відправляє дані без перезавантаження сторінки.

Анімації та ефекти CSS/JS забезпечують візуальний відгук на дії користувача: наприклад, завдяки псевдокласам `:hover` та `:active` [28] кнопки змінюють колір при наведенні, а зсув елемента нагору реалізується через `transform: translateY(-2px)` [7]. Використання CSS-змінних (`var(--accent-color)`) [14] дозволяє централізовано керувати колірною палітрою інтерфейсу.

У наступному блоці коду реалізовано стани кнопок (звичайний/`hover`) та їх плавне зміщення за допомогою `transform` і `transition` (рис. 3.31).

```
.checkout-btn:hover {
  background: var(--accent-color);
  transform: translateY(-2px);
}
```

Рис. 3.31 Ефект зсуву та зміни кольору кнопки при наведенні

На рис. 3.32-3.34 наведено приклади застосування transition з різними параметрами.

```
.btn {
  display: inline-block;
  padding: 1rem 2rem;
  background-color: var(--primary-color);
  color: white;
  text-decoration: none;
  border-radius: 5px;
  transition: background-color 0.3s ease;
  border: none;
  cursor: pointer;
}
```

Рис 3.32 Плавні переходи для кнопок

```
.quantity-btn {
  background: var(--primary-color);
  color: white;
  border: none;
  width: 24px;
  height: 24px;
  border-radius: 4px;
  cursor: pointer;
  display: flex;
  align-items: center;
  justify-content: center;
  font-size: 1rem;
  transition: background-color 0.2s ease;
}
```

Рис 3.33 Плавні переходи для кнопок в кошику

```
.checkout-btn {
  width: 100%;
  padding: 1rem;
  font-size: 1.1rem;
  display: flex;
  justify-content: center;
  align-items: center;
  gap: 0.8rem;
  margin-top: 1rem;
  background: var(--primary-color);
  color: white;
  border: none;
  border-radius: 8px;
  cursor: pointer;
  transition: all 0.3s ease;
}
```

Рис 3.34 Плавні переходи для кнопок оформлення замовлення

- transition: color 0.3s ease – плавна зміна кольору за 0.3 секунди
- transition: background-color 0.3s ease – плавна зміна фонового кольору за 0.3 секунди
- transition: transform 0.3s ease – плавна зміна трансформації за 0.3 секунди
- transition: all 0.3s ease – плавна зміна всіх властивостей за 0.3 секунди
- Параметр ease вказує на тип анімації – плавне прискорення та уповільнення.

Також при наведенні курсора на картку страви застосовується ефект підняття – картка зміщується трохи вгору (`translateY(-5px)`) та з'являється тінь навколо елемента (`box-shadow: 0 6px 12px rgba(0, 0, 0, 0.15)`) [6]. Це допомагає зробити елемент більш помітним серед інших (рис. 3.35).

```
.menu-item {  
  background: white;  
  border-radius: 10px;  
  overflow: hidden;  
  box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);  
  transition: transform 0.3s ease;  
}  
  
.menu-item:hover {  
  transform: translateY(-5px);  
}
```

Рис 3.35 Ефект підняття при наведенні курсора на карту та поява тіней

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було досягнуто поставленої мети – розроблено сучасний односторінковий вебсайт “Strelik” для інформування клієнтів, онлайн-бронювання столиків та замовлення страв, а також вирішено всі визначені завдання.

Виконано ґрунтовний аналіз історичного розвитку Інтернету та вебтехнологій: від ідеї Мемексу як першого концептуального прототипу асоціативного пошуку інформації, через винайдення гіпертексту й ARPANET, до сучасних стандартів HTML5, CSS3 та JavaScript. Такий огляд дозволив чітко зрозуміти еволюцію основних принципів побудови вебресурсів і закласти міцну методологічну базу для вибору технологічного стеку при розробці “Strelik”.

Проведено порівняльний аналіз функціональності та UX/UI-рішень трьох провідних українських мереж: “Пузата Хата”, “Львівські Круасани” та “Сушія”. Виявлено, що їхні сильні сторони це зрозуміла навігація, адаптивність та наявність програм лояльності. Разом з тим, виявлено і недоліки, зокрема відсутністю персоналізованого кабінету користувача й перевантаженістю інтерфейсу зайвими елементами. На основі цього аналізу сформульовано чіткі рекомендації щодо структури лендінгу, вибору ключових UI-патернів і оптимального набору функціональних блоків.

Розроблено інтерактивний макет із усіма ключовими екранами: головна сторінка з банером, динамічне меню з фільтрацією, форми бронювання та онлайн-замовлення, кошик із підрахунком вартості, блок контактів і футер. Проведене внутрішнє UX-тестування дозволило уточнити розташування елементів та поліпшити читабельність інтерфейсу.

Реалізовано семантичну розмітку із використанням сучасних тегів (<header>, <section>, <footer>), а також каскадні таблиці стилів із медіазапитами на точках перелому 480 px і 768 px. Застосовано відносні одиниці (% , vw, em) для відступів та шрифтів, що забезпечило коректне відображення на смартфонах, планшетах і десктопах.

Впроваджено безперезавантажувальну фільтрацію меню та клієнтську валідацію форм бронювання й замовлення з відображенням підтвердження успішних операцій. Це забезпечило плавну взаємодію користувача з сайтом у реальному часі та зменшило кількість помилок при введенні даних.

Використано ARIA-атрибути та семантичні теги для покращення доступності сайту людьми з обмеженими можливостями. Мультимедійні ресурси оптимізовано через lazy-loading та мінімізацію CSS/JS-файлів, що дозволило досягти часу першого повного завантаження сторінки менше ніж 2 с та високих балів у Google PageSpeed Insights.

Для подальшого розвитку сайту рекомендовано: розробити особистий кабінет користувача, в який додати модуль реєстрації, де зберігатиметься історія замовлень, уподобання та адреси доставки, що дозволить персоналізувати пропозиції та програми лояльності; доповнити аналітикою та CRM-інтеграцією, для цього впровадити збір статистики (Google Analytics, внутрішня панель) для відстеження ключових показників таких як популярність страв, час бронювання, географія замовлень; додати мультимовний інтерфейс, який реалізує локалізацію на кілька мов із автоматичним визначенням мови користувача та перемикачем мануальної адаптації, що розширить цільову аудиторію закладу; інтегрувати чат-бот, шляхом додавання модуля миттєвого спілкування для швидких відповідей на питання, що підвищить оперативність обслуговування.

Отримані результати свідчать про ефективність комплексного підходу: від глибокого теоретичного аналізу й огляду конкурентного середовища до практичної реалізації вебсайту з акцентом на UX/UI-дизайн, адаптивність і динамічний контент. Розроблений прототип “Strelik” демонструє високий рівень зручності користування (юзабіліті), швидкодію й доступність, і може слугувати шаблоном для інших закладів ресторанного бізнесу.

Готовий вебресурс та набір технічних рішень можуть бути впроваджені в роботу будь-якого ресторану для підвищення конверсії онлайн-бронювань, оптимізації обробки замовлень і зміцнення бренду. Рекомендації щодо поліпшення та розширення функціоналу можуть використовувати ІТ-команди або фрілансери при створенні схожих односторінкових лендінгів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. <button>: The Button element. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML/Reference/Elements/button> (дата звернення 20.04.2023).
2. ARPANET. URL: <https://surli.cc/olokls> (дата звернення 11.03.2023).
3. Array.prototype.filter description. URL: https://developer.mozilla.org/uk/docs/Web/JavaScript/Reference/Global_Objects/Array/filter (дата звернення 20.04.2023).
4. Array.prototype.filter(). URL: https://developer.mozilla.org/uk/docs/Web/JavaScript/Reference/Global_Objects/Array/filter (дата звернення 20.04.2023).
5. Array.prototype.map. URL: https://developer.mozilla.org/uk/docs/Web/JavaScript/Reference/Global_Objects/Array/map (дата звернення 20.04.2023).
6. Beautiful CSS box-shadow examples. URL: <https://getcssscan.com/css-box-shadow-examples> (дата звернення 20.04.2023).
7. CSS Function – translateY(). URL: https://www.tutorialspoint.com/css/css_func-translateY.htm (дата звернення 20.04.2023).
8. CSS transform. URL: <https://developer.mozilla.org/uk/docs/Web/CSS/transform> (дата звернення 20.04.2023).
9. Document.querySelector URL: <https://developer.mozilla.org/uk/docs/Web/API/Document/querySelectorAll> (дата звернення 20.04.2023).
10. Element.innerHTML property. URL: <https://developer.mozilla.org/uk/docs/Web/API/Element/innerHTML> (дата звернення 20.04.2023).
11. EventTarget: addEventListener() method. URL: <https://developer.mozilla.org/uk/docs/Web/API/EventTarget/addEventListener> (дата звернення 20.04.2023).

12. FormData. URL: <https://developer.mozilla.org/uk/docs/Web/API/FormData> (дата звернення 20.04.2023).

13. Honcharenko D. Як збільшити швидкість завантаження сторінок сайту на WordPress за допомогою Google PageSpeed Insights. URL: <https://hostpro.ua/blog/ua/how-to-increase-page-loading-speed-with-google-pagespeed-insights/> (дата звернення 17.03.2023).

14. How to use CSS variables like a pro. URL: <https://blog.logrocket.com/how-to-use-css-variables/> (дата звернення 20.04.2023).

15. HTML attribute maxlength. URL: <https://developer.mozilla.org/uk/docs/Web/HTML/Element/input#maxlength> (дата звернення 20.04.2023).

16. HTML attribute pattern. URL: <https://developer.mozilla.org/uk/docs/Web/HTML/Element/input#pattern> (дата звернення 20.04.2023).

17. HTML attribute type. URL: <https://developer.mozilla.org/uk/docs/Web/HTML/Element/input#type> (дата звернення 20.04.2023).

18. HTML URL: <https://uk.wikipedia.org/wiki/HTML> (дата звернення 20.04.2023).

19. HTMLElement: dataset property. URL: <https://developer.mozilla.org/uk/docs/Web/API/HTMLElement/dataset> (дата звернення 20.04.2023).

20. HTMLFormElement.reset() method. URL: <https://developer.mozilla.org/uk/docs/Web/API/HTMLFormElement/reset> (дата звернення 20.04.2023).

21. LVIV CROISSANTS. URL: <https://lvivcroissants.com/ua/> (дата звернення 17.03.2023).

22. Media Queries for Standard Devices. URL: <https://css-tricks.com/snippets/css/media-queries-for-standard-devices/> (дата звернення 20.04.2023).

23. Restaurant Website Builders: A Comprehensive Guide. URL: https://thedigitalrestaurant.com/restaurant-website-builders/?utm_source (дата звернення 17.03.2023).

24. Sara Cope. :required. URL: <https://css-tricks.com/almanac/pseudo-selectors/r/required/> (дата звернення 20.04.2023).

25. The Ultimate Guide to Responsive Website Design. URL: <https://www.hostpapa.com/blog/marketing/ultimate-guide-responsive-website-design-build-website-usability-mind/> (дата звернення 17.03.2023).

26. Using media queries. URL: https://developer.mozilla.org/uk/docs/Web/CSS/Media_Queries/Using_media_queries (дата звернення 20.04.2023).

27. Using responsive images in HTML. URL: https://developer.mozilla.org/uk/docs/Learn/HTML/Multimedia_and_embedding/Responsive_images (дата звернення 20.04.2023).

28. When do the :hover, :focus, and :active pseudo-classes apply? URL: <https://bitsofco.de/when-do-the-hover-focus-and-active-pseudo-classes-apply/> (дата звернення 20.04.2023).

29. Window: alert() method. URL: <https://developer.mozilla.org/uk/docs/Web/API/Window/alert> (дата звернення 20.04.2023).

30. World Wide Web. URL: <http://info.cern.ch/hypertext/WWW/TheProject.html> (дата звернення 11.03.2023).

31. Гіперпосилання HTML-сторінки. URL: <https://surl.li/pnhnfa> (дата звернення 11.03.2023).

32. Кобилянський І. Що таке лендинг пейдж? URL: <https://impulse-design.com.ua/ua/chto-takoe-landing-page.html> (дата звернення 20.04.2023).

33. Мемекс був задуманий у 1945 році? URL: <https://bayou.kultura.cx.ua/maysternist/memeks-buv-zadumaniy-u-1945-roci.html> (дата звернення 11.03.2023).

34. Мова розмітки даних SGML. URL: <https://surl.li/lklvpo> (дата звернення 11.03.2023).

35. Мова розмітки даних. URL: <https://surl.li/zznlvc> (дата звернення 11.03.2023).
36. Основні принципи UX-дизайну в продуктовому менеджменті. URL: <https://www.orbita.school/post/osnovni-principi-ux-dizaynu-v-produktovomu-menedzhmenti> (дата звернення 17.03.2023).
37. Пузата Хата. URL: <https://puzatahata.ua/> (дата звернення 17.03.2023).
38. Сушія. URL: <https://sushiya.ua/> (дата звернення 17.03.2023).
39. Функціональні та нефункціональні вимоги (з прикладами). URL: <https://surl.lu/nzbwoh> (дата звернення 11.03.2023).
40. Чому варто вивчати JavaScript? URL: <https://qagroup.com.ua/publications/why-learn-javascript/> (дата звернення 11.03.2023).
41. Що таке CSS. URL: https://css.in.ua/article/shcho-take-html_10 (дата звернення 20.04.2023).
42. Що таке JavaScript. URL: <https://cases.media/en/article/sho-take-javascript> (дата звернення 11.03.2023).
43. Що таке Концепція Memex? URL: <https://vukho.home.cx.ua/maysternist/shho-take-konceptiya-memex.html> (дата звернення 11.03.2023).