

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Бакалавр

(назва освітнього ступеня)

на тему: Розробка веб-сайту ресторану «Маяк»

Виконав: студент

IV курсу, групи СН-41

спеціальності

122 Комп'ютерні науки

(шифр і назва спеціальності)

		Бенедикт А.Г.
	(підпис)	(прізвище та ініціали)
Керівник		Липак Г.І.
	(підпис)	(прізвище та ініціали)
Нормоконтроль		Липак Г.І.
	(підпис)	(прізвище та ініціали)
Завідувач кафедри		Боднарчук І.О.
	(підпис)	(прізвище та ініціали)
Рецензент		
	(підпис)	(прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

« 27 » « січня » 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Бенедикту Артуру Георгійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка веб-сайту ресторану «Маяк»

Керівник роботи Липак Галина Ігорівна, доцент кафедри КН
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 07 » « травня » 2025 року № 4/7-444

2. Термін подання студентом завершеної роботи 23 червня 2025 р.

3. Вихідні дані до роботи Літературні та інтернет джерела щодо веб-технологій розроблення інтерактивних засобів взаємодії з користувачами

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз предметної області та формування вимог до веб-сайту «Маяк». 1.1 Загальна характеристика та актуальність веб-сайту «Маяк». 1.2 Аналіз існуючих аналогів до веб-сайту «Маяк». 1.3 Формування вимог до веб-сайту «Маяк». 1.4 Вибір інструментів для створення веб-сайту. 1.5 Застосування можливостей веб-сайту «Маяк». 1.6 Висновок до першого розділу. 2. Проектування основних елементів веб-сайту «Маяк». 2.1 Загальна архітектурна система веб-сайту «Маяк». 2.2 Проектування користувацького інтерфейсу веб-сайту «Маяк». 2.3 Забезпечення безпеки та захисту даних веб-сайту «Маяк». 2.4 Адаптивність та UX-дизайн веб-сайту «Маяк». 2.5 Управління замовленнями на веб-сайті ресторану «Маяк». 2.6 Висновок до другого розділу. 3. Тестування та введення у дію веб-сайту «Маяк». 3.1 Створена головна сторінка веб-сайту «Маяк». 3.2 Реалізований функціонал особистого кабінету користувача на веб-сайті ресторану «Маяк». 3.3 Введення веб-сайту в експлуатацію. 3.4 Висновок до третього розділу. 4. Безпека життєдіяльності, основи охорони праці. 4.1 Збереження емоційного здоров'я та уникнення професійного виснаження під час роботи за комп'ютером. 4.2 Комфортні умови до робочого місця оператора ПК. 4.3 Висновок до четвертого розділу. Висновки. Перелік Джерел. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Титульна сторінка. 2. Тема, мета, предмет дослідження. 3. Технічні засоби. 4. Актори веб-сайту «Маяк». 5. Спроектована архітектура. 6. Адаптивний UX-дизайн. 7. Головна сторінка веб-сайту. 8. Форма для входу користувачів. 9. Основна частина розділу «Наше меню». 10. Онлайн-форма бронювання на веб-сайті. 11. Головна сторінка особистого кабінету користувача. 12. Редагування профіля користувача. 13. Перегляд історії замовлень у особистому кабінеті. 15. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Сенчишин В.С., кандидат технічних наук, доцент кафедри МТ	02.06.2025	05.06.2025

7. Дата видачі завдання 27 січня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	27.01.2025	Виконано
2.	Підбір та опрацювання літературних джерел по темі кваліфікаційної роботи	31.01.2025-03.02.2025	Виконано
3.	Виконання дослідження щодо існуючих інструментів та технологій для розробки веб-сайтів	07.02.2025-09.02.2025	Виконано
4.	Розроблення структури та функціоналу веб-сайту «Маяк»	20.05.2025-01.06.2025	Виконано
5.	Оформлення розділу «Аналіз предметної області та формування вимог до веб-сайту ‘Маяк’»	02.06.2025-05.06.2025	Виконано
6.	Оформлення розділу «Проектування основних елементів веб-сайту ‘Маяк’»	06.06.2025-08.06.2025	Виконано
7.	Оформлення розділу «Тестування та введення у дію веб-сайту ‘Маяк’»	09.06.2025-11.06.2025	Виконано
8.	Виконання завдання до підрозділу «Безпека життєдіяльності»	12.06.2025-13.06.2025	Виконано
9.	Виконання завдання до підрозділу «Основи охорони праці»	14.06.2025-15.06.2025	Виконано
10.	Оформлення кваліфікаційної роботи	16.06.2025-17.06.2025	Виконано
11.	Нормоконтроль	18.06.2025-19.06.2025	Виконано
12.	Перевірка на плагіат	20.06.2025	Виконано
13.	Попередній захист кваліфікаційної роботи	21.06.2025	Виконано
14.	Захист кваліфікаційної роботи	24.06.2025	

Студент

(підпис)

Бенедикт А.Г.

(прізвище та ініціали)

Керівник роботи

(підпис)

Липак Г.І.

(прізвище та ініціали)

АНОТАЦІЯ

Розробка веб-сайту ресторану «Маяк» // Кваліфікаційна робота освітнього рівня «Бакалавр» // Бенедикт Артур Георгійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СН-41 // Тернопіль, 2025 // С. 66, рис. – 16, табл. – 2, кресл. – 0, додат. – 8, бібліогр. – 35.

Ключові слова: веб-сайт, ресторан, FastAPI, React, MongoDB, аутентифікація, замовлення, інтерфейс.

Кваліфікаційна робота присвячена розробці веб-сайту ресторану «Маяк» для покращення взаємодії з клієнтами та підвищення якості обслуговування.

У першому розділі кваліфікаційної роботи розглянуто аналіз предметної області, визначено основні функціональні вимоги та ключові об'єкти системи, а також побудовано діаграми варіантів використання веб-сайту «Маяк».

У другому розділі кваліфікаційної роботи проведено моделювання архітектури веб-сайту, проектування та практичну реалізацію структури як окремих функціональних компонентів, так і загальної структури веб-сайту.

У третьому розділі кваліфікаційної роботи детально розглянуто процес тестування та введення в експлуатацію веб-сайту «Маяк». Проведено аналіз отриманих результатів тестування, а також проаналізовано питання забезпечення безпеки та підвищення продуктивності веб-ресурсу.

ANNOTATION

Development of the restaurant website «Mayak» // Qualification work of the educational level «Bachelor» // Benedykt Artur Georgiyovich // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SN-41 // Ternopil, 2025 // P.66, fig. – 16, tabl. – 2, chair. – 0, annexes. – 8, references – 35.

Keywords: website, restaurant, FastAPI, React, MongoDB, authentication, orders, interface.

The qualification work is dedicated to the development of the «Mayak» restaurant website to improve customer interaction and service quality.

The first section of the qualification work discusses the analysis of the subject area, formulates a list of requirements and entities, and develops use case diagrams for the «Mayak» website.

The second section of the qualification work involves modeling the website's architecture, designing, and practical implementation of both individual functional components and the overall structure of the website.

The third section of the qualification work describes the process of testing and operating the «Mayak» website, analyzes the test results, and examines aspects of ensuring the security and the performance of the website.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

API (Application Programming Interface) – інтерфейс прикладного програмування, що дозволяє програмам взаємодіяти між собою.

Backend – серверна частина веб-додатку, що відповідає за бізнес-логіку та взаємодію з базою даних.

Frontend – клієнтська частина веб-додатку, що відповідає за інтерфейс користувача.

React – JavaScript-бібліотека для створення користувацьких інтерфейсів.

FastAPI – високопродуктивний веб-фреймворк для створення RESTful API на мові програмування Python.

JWT (JSON Web Token) – стандарт токенів для створення безпечного доступу до ресурсів веб-додатку.

SPA (Single Page Application) – односторінковий веб-застосунок, що завантажує єдину HTML-сторінку і динамічно оновлює її без повного перезавантаження.

MongoDB – документоорієнтована система керування базами даних, яка використовує формат зберігання JSON-подібних документів.

UX (User Experience) – користувацький досвід, що охоплює загальні враження користувача від взаємодії з продуктом.

UI (User Interface) – користувацький інтерфейс, візуальна частина веб-додатку, з якою безпосередньо взаємодіє користувач.

HTTPS (Hypertext Transfer Protocol Secure) – протокол передачі даних з підтримкою шифрування для забезпечення безпечного з'єднання.

CORS (Cross-Origin Resource Sharing) – механізм, що дозволяє виконувати крос-доменні запити між браузером і сервером.

SSL (Secure Sockets Layer) – протокол захисту даних, що забезпечує безпечну передачу інформації в мережі.

ORM (Object-Relational Mapping) – технологія, що забезпечує зв'язок між об'єктно-орієнтованими мовами програмування та реляційними базами даних.

REST API (Representational State Transfer API) – архітектурний стиль взаємодії компонентів мережі через HTTP-протокол.

CI/CD (Continuous Integration / Continuous Deployment) – методології безперервної інтеграції та розгортання програмного забезпечення.

UX/UI-дизайн – проектування взаємодії та інтерфейсу користувача, спрямоване на забезпечення зручності та доступності продукту.

HoReCa (Hotel, Restaurant, Cafe) – термін, що позначає індустрію гостинності та громадського харчування.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ФОРМУВАННЯ ВИМОГ ДО ВЕБ-САЙТУ «МАЯК»	11
1.1 Загальна характеристика та актуальність веб-сайту «Маяк»	11
1.2 Аналіз існуючих аналогів веб-сайтів «Маяк»	12
1.3 Формування вимог до веб-сайту «Маяк»	13
1.4 Вибір інструментів для створення веб-сайту «Маяк»	16
1.5 Застосування можливостей веб-сайту «Маяк»	18
1.6 Висновок до першого розділу	20
РОЗДІЛ 2. ПРОЕКТУВАННЯ ОСНОВНИХ ЕЛЕМЕНТІВ ВЕБ-САЙТУ «МАЯК»	22
2.1 Загальна архітектурна система веб-сайту «Маяк»	22
2.2 Проектування користувацького інтерфейсу веб-сайту «Маяк»	24
2.3 Забезпечення безпеки та захисту даних веб-сайту «Маяк»	27
2.4 Адаптивність та UX-дизайн веб-сайту «Маяк»	30
2.5 Управління замовленнями на веб-сайті ресторану «Маяк»	33
2.6 Висновок до другого розділу	35
РОЗДІЛ 3. ТЕСТУВАННЯ ТА ВВЕДЕННЯ У ДІЮ ВЕБ-САЙТУ «МАЯК»	38
3.1 Створена головна сторінка веб-сайту «Маяк»	38
3.2 Реалізований функціонал особистого кабінету користувача на веб-сайті ресторану «Маяк»	44
3.3 Введення веб-сайту в експлуатацію веб-сайту «Маяк»	49
3.4 Висновок до третього розділу	52
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	54
4.1 Працездатність людини – оператора	54
4.2 Затрати на заходи щодо покращення умов та охорони праці	56
4.3 Висновок до четвертого розділу	60
ВИСНОВКИ	61

ПЕРЕЛІК ДЖЕРЕЛ	63
ДОДАТКИ	

ВСТУП

Предмет дослідження. Вивчення методів та засобів створення інтерактивних веб-платформ, призначених для підвищення рівня сервісу у ресторанах.

Мета роботи. Покращення сервісу для відвідувачів ресторану шляхом впровадження сучасного, зручного та функціонального веб-сайту.

Актуальність дослідження. Сьогодні ресторанний бізнес стикається з високою конкуренцією та необхідністю відповідати очікуванням клієнтів, які цінують швидкість, зручність і сучасні технології. Використання веб-ресурсів, які дозволяють здійснювати онлайн-замовлення, бронювати столи, читати новини закладу чи спілкуватися із персоналом, стало невід'ємною частиною розвитку сучасних ресторанів. Створення веб-сайту для ресторану «Маяк» цілком відповідає сучасним тенденціям розвитку ІТ-сфери та є актуальним напрямком для досліджень і впроваджень.

Мета і завдання дослідження. Головною метою цієї бакалаврської роботи є підвищення якості обслуговування у ресторані «Маяк» через створення ефективного веб-сайту. Для досягнення цієї мети необхідно:

- Дослідити сучасний стан і новітні технології розробки веб-сайтів для ресторанів.
- Побудувати структуру веб-сайту, передбачивши можливості онлайн-замовлень, резервування столиків, ведення блогу та управління профілями користувачів.
- Реалізувати систему авторизації та реєстрації з поділом прав для адміністрації, кухарів та клієнтів.
- Створити адаптивний та ергономічний дизайн для зручності користування.
- Перевірити працездатність, продуктивність і безпечність веб-сайту за допомогою відповідного тестування

Практичне значення результатів. Розроблений у межах дослідження веб-сайт дозволить ресторану «Маяк» не лише підвищити якість обслуговування й кількість онлайн-замовлень та бронювань, а й активніше комунікувати з гостями через інтерактивний блог. Це стане дієвим інструментом для просування ресторану в мережі Інтернет та підвищить його конкурентоспроможність на ринку.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ФОРМУВАННЯ ВИМОГ ВЕБ-САЙТУ «МАЯК»

1.1 Загальна характеристика та актуальність веб-сайту «Маяк»

У сучасних умовах розвитку цифрових технологій наявність веб-сайту для закладів громадського харчування є не лише перевагою, а й необхідністю. Веб-сайт виконує роль основного інформаційного ресурсу, за допомогою якого клієнти можуть ознайомитися з меню, атмосферою ресторану, акційними пропозиціями, а також здійснити попереднє бронювання столика або оформити замовлення онлайн [1].

Ресторан «Маяк» позиціонується як сучасний заклад з якісним сервісом та унікальною концепцією. Для підтримання конкурентоспроможності на ринку необхідно забезпечити зручний і функціональний канал комунікації з клієнтами. Саме веб-сайт здатен стати ефективним інструментом для презентації ресторану в онлайн-просторі, підвищення впізнаваності бренду та залучення нових відвідувачів.

Актуальність створення веб-сайту полягає також у зміні поведінки споживачів: більшість користувачів перед відвідуванням закладу шукають інформацію про нього в Інтернеті. Відсутність власного сайту або його застарілий вигляд може створити враження непрофесійності, тоді як сучасний, адаптивний вебресурс з інтуїтивно зрозумілим інтерфейсом і актуальним контентом, навпаки, викликає довіру.

Запровадження веб-сайту дозволить автоматизувати частину процесів: онлайн-бронювання, перегляд меню, зворотний зв'язок із клієнтами. Це не лише підвищить якість обслуговування, а й зменшить навантаження на персонал закладу.

Таким чином, розробка веб-сайту ресторану «Маяк» є актуальним та доцільним кроком, спрямованим на покращення сервісу, розширення аудиторії та цифрову трансформацію бізнесу.

1.2 Аналіз існуючих аналогів веб-сайтів «Маяк»

Для розробки якісного та конкурентоспроможного веб-сайту ресторану «Маяк» важливо провести детальний аналіз існуючих аналогів — веб-ресурсів ресторанів, які мають схожий формат і цільову аудиторію. Такий аналіз допомагає виявити сучасні тенденції в дизайні, функціональності, структурі та користувацькому досвіді, а також визначити сильні та слабкі сторони, які варто врахувати при створенні власного сайту.

За результатами огляду найкращих веб-сайтів ресторанної сфери, серед яких сайти зібрані у статті «17 прикладів найкращих сайтів ресторанів та кафе» на сайті Impulse Design, можна виокремити кілька ключових аспектів, що роблять сайт успішним.

Адаптивний та стильний дизайн — це фундамент будь-якого сучасного сайту. Веб-ресурси, які демонструють високий рівень залучення користувачів, мають дизайн, що відповідає стилю та атмосфері закладу. Такий дизайн не лише підкреслює унікальність бренду, а й забезпечує комфортний перегляд на різних пристроях — від смартфонів до настільних комп'ютерів. Адаптивність сайту дозволяє зберігати функціональність і привабливість інтерфейсу, незалежно від розміру екрана, що є критично важливим для сучасного мобільного користувача.

Зручна і логічна навігація — ще один важливий компонент, який допомагає користувачам швидко та без зусиль знаходити необхідну інформацію. Меню має бути структурованим і зрозумілим, із чіткими категоріями та підрозділами. Легкий доступ до основних розділів, таких як «Меню», «Про нас», «Контакти» та «Онлайн-бронювання», значно підвищує зручність користування сайтом та позитивно впливає на загальне враження.

Інтеграція з онлайн-замовленнями та платіжними системами є критичною для сучасних ресторанів, особливо в умовах зростаючого попиту на дистанційні сервіси. Ефективний веб-сайт повинен забезпечувати користувачам можливість легко і безпечно оформити замовлення та оплатити його через зручні платіжні

інструменти. Такий функціонал не лише покращує клієнтський досвід, а й підвищує операційну ефективність закладу.

Наявність розділів з меню, фотогалереєю, контактами та онлайн-бронюванням дозволяє максимально повно представити заклад, його атмосферу та пропозиції. Фотогалерея виконує роль візуального маркетингу, допомагаючи створити позитивне враження ще до фізичного візиту. Розділ з контактами та інтеграція з картами дозволяють легко знайти ресторан, а онлайн-бронювання зменшує час очікування і підвищує зручність [2].

Оптимізація швидкості завантаження сайту також є ключовою складовою, адже швидкість впливає на користувацький досвід та SEO-рейтинги. Сайти, що повільно завантажуються, ризикують втратити відвідувачів і, відповідно, потенційних клієнтів. Впровадження сучасних технологій оптимізації, таких як кешування, стиснення зображень і мінімізація коду, допомагає забезпечити швидку та плавну роботу ресурсу.

1.3 Формування вимог до веб-сайту «Маяк»

Вимоги до веб-сайту ресторану «Маяк» спрямовані на забезпечення високого рівня обслуговування клієнтів та зручності користування ресурсом. Вони поділяються на функціональні, нефункціональні, а також включають вимоги щодо інтеграції з іншими системами. Функціональні вимоги визначають перелік основних можливостей, якими повинен володіти сайт, зокрема: перегляд меню в інтерактивному вигляді, оформлення та відстеження замовлень, реєстрація та вхід користувачів, можливість редагування персональної інформації в особистому кабінеті, система бронювання столиків онлайн, а також підтримка комунікації з адміністрацією ресторану через зворотний зв'язок або чат.

Нефункціональні вимоги стосуються якості та зручності використання веб-ресурсу, стабільної роботи при великій кількості відвідувачів, швидкого завантаження сторінок, адаптивності інтерфейсу для різних типів пристроїв,

безпеки збереження та обробки персональних даних користувачів, регулярного резервного копіювання інформації, багатомовності та доступності для людей з різними рівнями цифрових навичок. Окрему увагу приділено питанням масштабованості, можливості розширення функціоналу у майбутньому та інтеграції з зовнішніми сервісами, такими як платіжні системи, служби доставки, соціальні мережі й аналітичні платформи.

Проектування архітектури сайту полягає у розподілі системи на окремі модулі, встановленні їхніх функцій і відповідальностей, а також способів взаємодії між ними. Такий підхід дозволить сформувати узгоджений план розробки, що гарантуватиме ефективне виконання як функціональних, так і нефункціональних вимог. Основні компоненти системи включатимуть: систему управління контентом, модулі для обробки замовлень та систему бронювання столиків.

Нижче наведено таблицю 1.1 з основними вимогами до веб-сайту «Маяк».

Таблиця 1.1 – Основні вимоги до веб-сайту «Маяк»

Категорія	Вимоги
1	2
Функціональні вимоги	- Відображення меню з детальним описом і фотографіями страв, можливість сортування за категоріями. - Онлайн-замовлення: додавання страв до кошика, оформлення замовлення, підтвердження по e-mail або SMS. - Резервування столиків через інтерактивну форму з можливістю підтвердження та скасування бронювання.

1	2
Нефункціональні вимоги	- Висока продуктивність: швидке завантаження сторінок, оптимізація зображень і медіа. - Безпека: захист персональних даних, використання SSL-сертифікатів, захист від CSRF та XSS-атак. - Зручність використання: інтуїтивно зрозумілий інтерфейс, адаптивний дизайн, проста навігація. - Надійність: забезпечення високої доступності, регулярне резервне копіювання.
Вимоги до інтеграції	- Інтеграція з платіжними системами: підтримка різних способів оплати, безпечне зберігання та обробка платіжних даних

Для створення ефективного веб-сайту необхідно розробити прототипи та макети основних сторінок, які допоможуть візуалізувати дизайн, гарантувати його цілісність, узгодженість та відповідність сучасним стандартам UX/UI. Важливо враховувати різні сценарії взаємодії користувачів із сайтом, протестувати логіку навігації між сторінками, а також адаптивність інтерфейсу для мобільних пристроїв і десктопів. Після затвердження макетів починається етап розробки функціоналу з поетапним впровадженням і детальним тестуванням усіх функцій веб-ресурсу [3].

Особлива увага приділяється забезпеченню високої продуктивності, захисту персональних даних користувачів, інформаційній безпеці та стабільності роботи системи під час значних навантажень. Важливим аспектом є інтеграція з зовнішніми сервісами, такими як платіжні системи, мапи, аналітичні інструменти та соціальні мережі, що дає змогу розширити можливості сайту й підвищити зручність для клієнтів. Крім того, необхідно реалізувати ефективну адміністративну панель для менеджменту контенту й

оперативного реагування на відгуки користувачів та замовлення. Постійний моніторинг працездатності сайту, впровадження систем логуювання і зворотного зв'язку дозволить швидко реагувати на потенційні проблеми та забезпечити стабільну роботу ресурсу.

Завдяки комплексному підходу до проектування, розробки та тестування веб-сайт ресторану «Маяк» максимально відповідатиме як очікуванням користувачів, так і бізнес-вимогам, сприятиме підвищенню лояльності клієнтів, ефективності роботи персоналу і стане дієвим інструментом для подальшого розвитку та просування закладу в сучасному конкурентному середовищі.

1.4 Вибір інструментів для створення веб-сайту

Для успішної розробки веб-сайту ресторану «Маяк» важливим етапом є правильний вибір технологій та інструментів, що забезпечать необхідну функціональність, продуктивність, безпеку, зручність подальшої підтримки та масштабування проекту. Компетентний підхід до вибору стеку технологій дає можливість створити ефективний сучасний веб-ресурс, який відповідатиме як потребам користувачів, так і бізнес-завданням закладу.

Фронтенд. Для створення сучасного користувацького інтерфейсу рекомендується використовувати передові JavaScript-фреймворки, такі як React, Vue.js або Angular. Вони дозволяють розробити динамічний, адаптивний та інтуїтивно зрозумілий інтерфейс, який швидко реагує на дії користувача, підтримує гнучку навігацію та забезпечує якісний користувацький досвід незалежно від пристрою. Для організації стилів і створення привабливого дизайну можуть використовуватись CSS-фреймворки, зокрема Bootstrap, Tailwind CSS або Material UI, що пришвидшує верстку і забезпечує кросбраузерну сумісність.

Бекенд. На стороні сервера доцільно застосувати технології, які гарантують стабільну роботу ресурсу, високу продуктивність, а також легку інтеграцію з різноманітними базами даних та зовнішніми сервісами. Це можуть

бути такі інструменти, як Node.js з фреймворком Express, сучасні Python-фреймворки (наприклад, Django чи FastAPI), або перевірені рішення на PHP (Laravel). Вибір серверної платформи повинен враховувати технічні навички команди, особливості проєкту, вимоги до безпеки та майбутнього масштабування системи.

База даних. Для зберігання інформації про меню, замовлення, бронювання, користувачів та інші бізнес-дані найкраще підійдуть реляційні бази даних — MySQL або PostgreSQL. Вони забезпечують високу надійність збереження інформації, підтримку складних аналітичних запитів, можливість створення резервних копій і масштабування за потреби. Для додаткових функцій, наприклад, кешування, можна використовувати Redis чи Memcache.

Інструменти розробки та управління. Для організації ефективної командної роботи та контролю версій доцільно впровадити Git у поєднанні з платформами для співпраці, наприклад, GitHub або GitLab. Для планування, моніторингу завдань і керування проєктом доцільно використовувати Jira, Trello чи Asana. Основним середовищем розробки (IDE) може бути Visual Studio Code, WebStorm або інші сучасні редактори коду, які забезпечують швидку навігацію по проєкту та інтеграцію з різними плагінами.

Інші важливі інструменти:

- Системи автоматичного тестування (наприклад, Jest, Mocha, Cypress для фронтенду, PyTest для бекенду) — допомагають підтримувати якість коду, виявляти помилки ще до релізу.
- CI/CD-платформи (наприклад, GitHub Actions, GitLab CI/CD, Jenkins) — автоматизують процеси тестування, збирання та розгортання додатку, що дозволяє скоротити час виходу оновлень на продакшн.
- Системи моніторингу та логування (Prometheus, Grafana, Sentry, ELK Stack) дають можливість відслідковувати стабільність роботи сайту, аналізувати помилки, швидко реагувати на можливі проблеми з безпекою чи продуктивністю.

– Інструменти для аудиту безпеки (наприклад, OWASP ZAP), а також впровадження SSL-сертифікатів для захищеного з'єднання та сучасних протоколів аутентифікації (наприклад, OAuth 2.0 або JWT).

Хмарні сервіси та розгортання. Для розміщення і масштабування веб-сайту можна використовувати сучасні хмарні платформи (AWS, Google Cloud Platform, Microsoft Azure), або спеціалізовані сервіси для розгортання додатків (Vercel, Netlify, Heroku).

Комплексне використання перелічених технологій і інструментів [4-6] дозволяє створити сучасний, безпечний, масштабований та зручний у підтримці веб-сайт ресторану «Маяк», що буде відповідати високим вимогам бізнесу та очікуванням користувачів, забезпечуючи їм якісний сервіс та приємний досвід взаємодії із закладом у цифровому просторі.

1.5 Застосування можливостей веб-сайту «Маяк»

Веб-сайт «Маяк» розроблений із використанням сучасних технологій для забезпечення максимальної зручності користувачів і підвищення ефективності роботи закладу. Його функціональні можливості реалізовані таким чином, щоб повністю відповідати потребам клієнтів та адміністрації ресторану.

Основні напрямки застосування функціоналу:

- Оформлення онлайн-замовлень. Відвідувачі можуть швидко і просто вибрати страви з меню, здійснити замовлення та оплатити його онлайн, що значно економить час як клієнтів, так і персоналу.
- Резервування столиків. Інтерфейс сайту дозволяє забронювати столик на зручний час без необхідності телефонного дзвінка, що підвищує комфорт гостей.
- Зворотній зв'язок та оцінки. Відвідувачі мають можливість залишати свої відгуки, побажання або звертатись із питаннями через контактну форму, що дозволяє адміністрації реагувати на потреби клієнтів.

- Інформаційна підтримка. Користувачі завжди можуть дізнатися про актуальні новини, акції та спеціальні пропозиції через блог або розділ новин. Незареєстрований користувач – це відвідувач сайту, який має доступ до загальнодоступних функцій, таких як перегляд меню, ознайомлення з інформацією про ресторан. Зареєстрований користувач – це відвідувач сайту, який створив обліковий запис і увійшов у систему, використовуючи свої облікові дані. Зареєстрований користувач має доступ до всіх функцій, доступних незареєстрованим користувачам, таких як перегляд меню, ознайомлення з інформацією про ресторан. Нижче наведено рисунок 1.1, що ілюструє основних акторів та їхні дії.

- Адміністрування та управління контентом. Для працівників ресторану реалізована окрема адміністративна панель, за допомогою якої можна оновлювати меню, переглядати замовлення, керувати новинами та реагувати на відгуки клієнтів.

Впровадження цих можливостей дозволяє підвищити якість сервісу, покращити комунікацію з відвідувачами та автоматизувати основні бізнес-процеси.

Аналіз предметної області та спеціалізованої літератури [7] дозволив визначити головних акторів проєктованого веб-сайту (рис. 1.1).

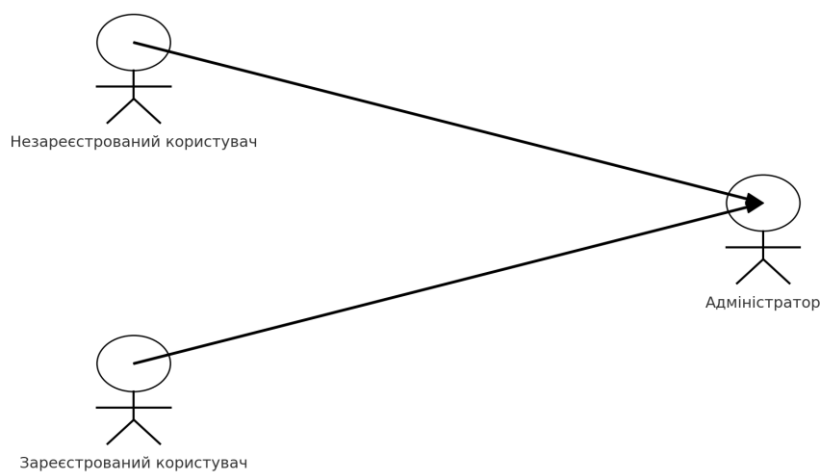


Рисунок 1.1 – Актори веб-сайту «Маяк»

1.6 Висновок до першого розділу

У першому розділі було проведено всебічний аналіз предметної області, визначено актуальність створення веб-сайту для ресторану «Маяк» та проаналізовано особливості змін у поведінці сучасних споживачів. Було доведено, що наявність сучасного, функціонального та адаптивного веб-ресурсу є ключовою умовою для підтримання конкурентоспроможності ресторану та підвищення рівня обслуговування клієнтів.

Проведений аналіз існуючих аналогів веб-сайтів ресторанів дозволив визначити сильні та слабкі сторони популярних рішень, виокремити основні функціональні блоки та елементи дизайну, що відповідають сучасним стандартам. На основі отриманих результатів було сформульовано комплексні вимоги до майбутнього веб-сайту «Маяк» — як функціональні, так і нефункціональні, а також вимоги до інтеграції з зовнішніми системами.

Окремо обґрунтовано вибір інструментів і технологій для розробки веб-ресурсу, враховуючи необхідність забезпечення продуктивності, безпеки, масштабованості та зручності подальшої підтримки проекту. Також було розглянуто основні можливості та сценарії застосування функціоналу сайту, що дозволяють оптимізувати бізнес-процеси та покращити комунікацію з клієнтами.

Одним із перспективних напрямів подальшої розробки є впровадження програми лояльності для постійних клієнтів. Вона передбачатиме надання індивідуальних знижок, бонусів та спеціальних пропозицій, що стимулюватиме клієнтів до повторного відвідування та сприятиме формуванню постійної аудиторії. Для розширення можливостей взаємодії з користувачами доцільним є також створення мобільного застосунку, який виступатиме функціональним продовженням веб-сайту. Такий застосунок забезпечить зручний доступ до сервісів ресторану з будь-яких пристроїв, підвищуючи рівень комфорту користувачів та покращуючи їхній досвід.

Крім того, важливо реалізувати систему зворотного зв'язку, що дозволить клієнтам залишати відгуки, оцінки та побажання. Це надасть адміністрації можливість оперативно реагувати на звернення, що сприятиме підвищенню якості обслуговування та формуванню позитивного іміджу закладу.

Загалом, результати першого розділу слугують надійною теоретичною та методологічною основою для розробки сучасного веб-сайту ресторану «Маяк», який повністю відповідатиме актуальним потребам як бізнесу, так і його цільової аудиторії.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ОСНОВНИХ ЕЛЕМЕНТІВ ВЕБ-САЙТУ «МАЯК»

2.1 Загальна архітектурна система веб-сайту «Маяк»

Архітектура веб-сайту «Маяк» базується на класичній клієнт-серверній моделі, яка забезпечує розподіл обов'язків між інтерфейсною (фронтенд) та серверною (бекенд) частинами, що дозволяє досягти гнучкості, масштабованості та високої продуктивності системи. Фронтенд частину застосунку реалізовано з використанням бібліотеки React, яка дає змогу створювати динамічний та високопродуктивний інтерфейс користувача, як показано на рисунку 2.1. Компоненти React відповідають за рендеринг сторінок, обробку дій користувача та взаємодію з бекендом через API-запити. Завдяки використанню React можлива реалізація односторінкових застосунків (SPA), що значно покращує користувацький досвід шляхом миттєвого оновлення контенту без необхідності перезавантаження сторінки.

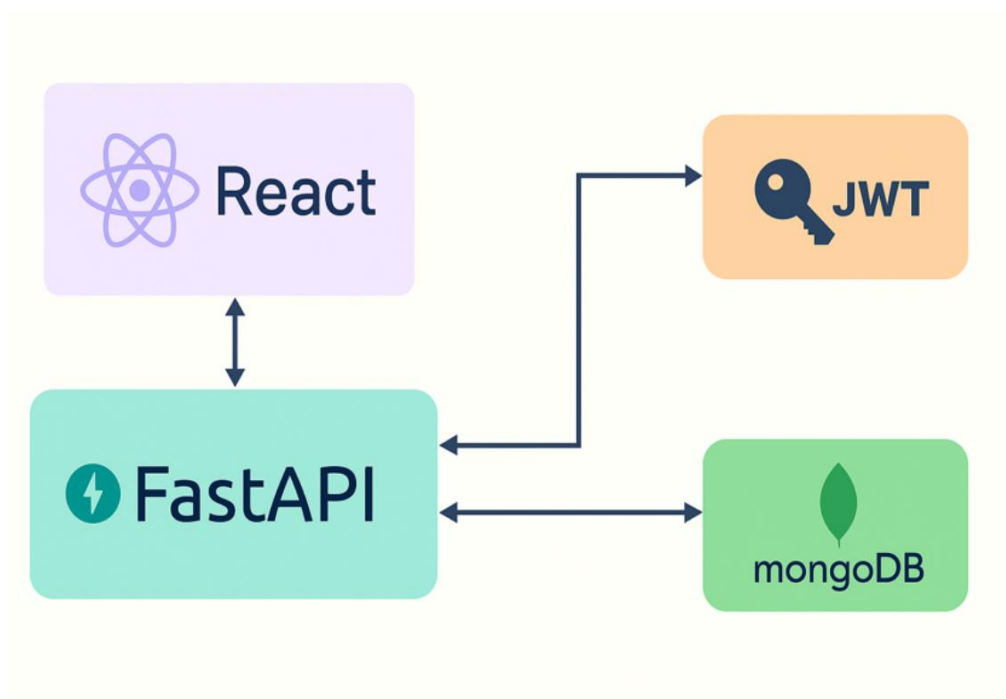


Рисунок 2.1 – Спроектована архітектура веб-сайту «Маяк»

Бекенд веб-сайту реалізовано на FastAPI — високопродуктивному фреймворку Python, що приймає HTTP-запити, виконує бізнес-логіку та взаємодіє з базою даних. Для автентифікації та авторизації застосовується JSON Web Token (JWT): клієнт зберігає токен локально й додає його до кожного запиту, тож сервер може надійно перевіряти права доступу та запобігати несанкціонованим операціям.

Основне сховище даних веб-платформи побудоване на базі MongoDB — сучасної документоорієнтованої системи управління базами даних, яка особливо ефективна для роботи з напівструктурованою інформацією, великими обсягами даних та гнучкою зміною структури. Вибір саме цієї СУБД зумовлений її здатністю масштабуватись у горизонтальному напрямку, високою швидкістю читання й запису, а також зручністю інтеграції з сучасними веб-технологіями. MongoDB дозволяє зберігати дані у вигляді колекцій документів (формату JSON/BSON), що значно спрощує організацію інформації, її пошук, фільтрацію та модифікацію у процесі експлуатації сайту [8].

Архітектура взаємодії всіх компонентів платформи «Маяк» складається з чотирьох основних елементів: фронтенду, бекенду, модуля аутентифікації (JWT) та бази даних MongoDB. На стороні клієнта (фронтенд) використовується React — потужний JavaScript-фреймворк, який забезпечує побудову динамічного та адаптивного користувацького інтерфейсу. Бекенд реалізовано на основі FastAPI — сучасного високопродуктивного Python-фреймворку, що дозволяє швидко створювати масштабовані RESTful-сервіси з підтримкою асинхронних операцій.

Модуль аутентифікації заснований на JSON Web Token (JWT), який забезпечує безпечну і зручну авторизацію користувачів, контроль доступу до захищених ресурсів та розмежування прав відповідно до ролей. При кожному запиті React-інтерфейс надсилає запит до бекенду FastAPI, який здійснює перевірку аутентифікації за допомогою JWT, після чого звертається до MongoDB для отримання або оновлення даних. Всі дані між клієнтом і

сервером передаються через захищений протокол HTTPS, що додатково підвищує рівень інформаційної безпеки.

Комунікація між компонентами здійснюється через REST API, що дає змогу легко масштабувати систему, інтегрувати нові сервіси або змінювати окремі модулі без істотного впливу на інші частини архітектури. Такий підхід дозволяє підтримувати гнучкість і модульність веб-платформи, полегшує впровадження нових функцій, оперативно оновлювати функціонал або оптимізувати його відповідно до зростаючих вимог користувачів та бізнесу.

Завдяки вибору стеку React + FastAPI + MongoDB із використанням JWT, платформа «Маяк» відповідає сучасним вимогам до продуктивності, безпеки, масштабованості та простоти підтримки, а також забезпечує можливість подальшого розвитку та інтеграції із зовнішніми сервісами та інструментами аналітики.

2.2 Проектування користувацького інтерфейсу веб-сайту «Маяк»

Проектування користувацького інтерфейсу (UI) є одним із ключових етапів розробки веб-сайту «Маяк», оскільки саме зовнішній вигляд і зручність взаємодії безпосередньо впливають на залученість відвідувачів і формування позитивного досвіду користувача (UX). Сучасний веб-сайт має не тільки відображати стиль і концепцію ресторану, а й відповідати високим вимогам до ергономіки, простоти та логічності взаємодії, що особливо важливо для першого враження та подальшої лояльності користувачів.

Під час розробки інтерфейсу, як зображено на рисунку 2.2, враховувалися сучасні принципи юзабіліті, доступності й адаптивності. Сайт має лаконічну, зрозумілу структуру, яка дозволяє відвідувачам легко орієнтуватися серед великої кількості інформації та швидко знаходити потрібні розділи. Головна сторінка містить коротку презентацію ресторану, основні переваги, фотогалерею, інтегрований блок із новинами та акціями. Важливу роль відіграє навігаційне меню, розташоване у верхній частині екрану, яке забезпечує

швидкий доступ до таких розділів, як «Про нас», «Меню», «Новини», «Контакти», «Послуги», «Вхід» та «Забронювати».

Дизайн сайту витриманий у сучасній кольоровій гамі, яка відповідає фірмовому стилю ресторану «Маяк», підкреслює його атмосферу та сприяє емоційному сприйняттю бренду. Використання візуальних елементів, таких як якісні фотографії страв, інтерактивні кнопки, іконки для дій, підсилює залученість і мотивує користувачів до подальших дій на сайті (наприклад, оформлення замовлення або бронювання столика).

Особливу увагу приділено адаптивності — інтерфейс коректно відображається на різних типах пристроїв: комп'ютерах, планшетах, смартфонах. Всі елементи автоматично змінюють розмір і розташування залежно від роздільної здатності екрану, що дозволяє користувачам комфортно взаємодіяти із сайтом у будь-яких умовах. Додатково передбачено режим підвищеної контрастності та підтримку масштабування шрифтів для людей із порушеннями зору.

У процесі проектування були використані wireframes (каркаси сторінок) та інтерактивні прототипи, що дало змогу ретельно опрацювати кожную деталь інтерфейсу, забезпечити цілісність дизайну й узгодженість між усіма сторінками сайту. Проведення тестування прототипів на різних користувачах дозволило виявити та усунути можливі незручності ще до етапу програмної реалізації.

Інтерфейс забезпечує не лише естетичну привабливість, а й високу швидкість взаємодії — користувачі можуть з легкістю здійснювати пошук страв, сортувати меню за категоріями, залишати відгуки, підписуватися на новини, переглядати фотогалерею та здійснювати онлайн-замовлення. Дружній до користувача дизайн сприяє зростанню рівня задоволеності клієнтів, повторному відвідуванню веб-сайту і формуванню позитивного іміджу ресторану «Маяк» в цифровому просторі.

Завдяки впровадженню сучасних принципів UI/UX, сайт стає ефективним інструментом комунікації з клієнтами, підтримує бізнес-процеси закладу й відповідає вимогам ринку HoReCa щодо цифрових сервісів.

Контакти	Послуги	Новини	Вхід
----------	---------	--------	------

Про нас	Меню	Забронювати
---------	------	-------------

Рисунок 2.2 – Основні розділи інтерфейсу веб-сайту «Маяк»

Для оформлення використовуються легкі кольори, читабельні шрифти та достатній простір між елементами для полегшення сприйняття. Всі інтерактивні елементи — кнопки, форми, посилання — виконані в єдиному стилі, що сприяє цілісності дизайну [16]. Інтерфейс адаптований для різних типів пристроїв, завдяки чому сайт однаково зручний як на комп'ютерах, так і на мобільних гаджетах.

Особливу увагу приділено зворотному зв'язку з користувачем: на сайті реалізовані спливаючі підказки, повідомлення про помилки та успішні дії. Для користувачів з вадами зору передбачено підтримку масштабування тексту й коректну роботу скрінрідерів.

Таким чином, проектування UI веб-сайту «Маяк» було спрямовано на забезпечення комфортного та доступного користування ресурсом для максимально широкої аудиторії.

2.3 Забезпечення безпеки та захисту даних веб-сайту «Маяк»

Забезпечення безпеки та захисту даних веб-сайту ресторану «Маяк» є одним із найважливіших завдань у процесі розробки та подальшої експлуатації ресурсу. Враховуючи сучасні кіберзагрози, особливу увагу приділено впровадженню комплексної системи безпеки для захисту персональної та фінансової інформації користувачів, а також стабільної роботи сайту загалом.

Для захисту даних користувачів застосовуються сучасні технології аутентифікації та авторизації, зокрема токени JSON Web Token (JWT), які дозволяють надійно перевіряти особу відвідувача та обмежувати доступ до конфіденційної інформації відповідно до ролей і прав [17]. Використання JWT дає змогу мінімізувати ризики несанкціонованого доступу до особистих кабінетів, панелі адміністратора та інших чутливих розділів сайту.

Уся інформація між клієнтом і сервером передається виключно через захищений протокол HTTPS, що унеможливорює перехоплення чи підміну даних зловмисниками. Для зберігання паролів використовуються надійні алгоритми хешування, наприклад, bcrypt, які забезпечують додатковий рівень захисту навіть у разі компрометації бази даних. Окрім того, персональні дані користувачів у базі можуть бути додатково зашифровані, щоб унеможливити їх використання сторонніми особами.

У процесі реєстрації користувач заповнює спеціальну форму, де обов'язково зазначає ім'я, прізвище, електронну адресу, номер телефону, пароль та підтвердження пароля. Для забезпечення відповідності чинному законодавству та захисту прав користувача, у формі передбачено галочку для підтвердження згоди на обробку персональних даних. Після внесення всіх необхідних відомостей та натискання кнопки «Зареєструватися», дані проходять попередню валідацію — перевіряється правильність формату електронної пошти, складність пароля, унікальність акаунту, що мінімізує кількість помилок і дублювань у системі [18].

Після успішної реєстрації та підтвердження електронної адреси (через надісланий лист із посиланням для активації акаунту) користувач отримує можливість авторизуватися та отримує доступ до особистого кабінету на сайті. У власному профілі користувач може змінювати персональні дані, переглядати історію замовлень, бронювання, залишати відгуки чи комунікувати з адміністрацією ресторану через захищену форму зворотного зв'язку. Усі операції, що стосуються редагування даних чи оплати, додатково супроводжуються перевіркою прав доступу та двофакторною аутентифікацією (за бажанням користувача), що підвищує рівень захисту облікового запису.

Уся інформація між клієнтом і сервером передається виключно через захищений протокол HTTPS, що унеможливорює перехоплення чи підміну даних зловмисниками. Для зберігання паролів використовуються надійні алгоритми хешування, наприклад, bcrypt, які забезпечують додатковий рівень захисту навіть у разі компрометації бази даних. Окрім того, персональні дані користувачів у базі можуть бути додатково зашифровані, щоб унеможливити їх використання сторонніми особами.

У процесі реєстрації користувач заповнює спеціальну форму, де обов'язково зазначає ім'я, прізвище, електронну адресу, номер телефону, пароль та підтвердження пароля. Для забезпечення відповідності чинному законодавству та захисту прав користувача, у формі передбачено галочку для підтвердження згоди на обробку персональних даних. Після внесення всіх необхідних відомостей та натискання кнопки «Зареєструватися», дані проходять попередню валідацію — перевіряється правильність формату електронної пошти, складність пароля, унікальність акаунту, що мінімізує кількість помилок і дублювань у системі.

Після успішної реєстрації та підтвердження електронної адреси (через надісланий лист із посиланням для активації акаунту) користувач отримує можливість авторизуватися та отримує доступ до особистого кабінету на сайті. У власному профілі користувач може змінювати персональні дані, переглядати історію замовлень, бронювання, залишати відгуки чи комунікувати з

адміністрацією ресторану через захищену форму зворотного зв'язку. Усі операції, що стосуються редагування даних чи оплати, додатково супроводжуються перевіркою прав доступу та двофакторною аутентифікацією (за бажанням користувача), що підвищує рівень захисту облікового запису.

Також системою впроваджено аудит дій користувачів та адміністраторів, ключові операції фіксуються у спеціальних логах, що дає змогу оперативно виявляти та реагувати на підозрілі активності чи потенційні загрози. Постійне оновлення програмного забезпечення, встановлення актуальних патчів безпеки, а також проведення регулярного аудиту вразливостей забезпечують надійний захист сайту ресторану «Маяк» від сучасних кіберзагроз.

Візуальний вигляд та структура форми реєстрації зображені на рисунку, що демонструє зрозумілий та зручний для користувачів інтерфейс, який сприяє швидкому й безпечному створенню облікового запису.

Рис. 2.3. Спроекований бланк форми реєстрації користувача веб-сайту «Маяк». Форму складає три елементи: поле для введення логіну, поле для введення паролю та кнопка «Відправити».

Рисунок 2.3 – Спроекований бланк форми реєстрації користувача веб-сайту «Маяк»

Особлива увага приділяється захисту від типових кіберзагроз: усі дані, що надходять від користувачів, проходять серверну валідацію та фільтрацію, впроваджено захист від атак типу XSS та CSRF. Для зменшення ризику несанкціонованого доступу адміністративна панель доступна лише з певних IP-адрес. Регулярно оновлюється програмне забезпечення, застосовуються сучасні

брандмауери для фільтрації шкідливого трафіку, а також ведеться постійний моніторинг і журналювання дій користувачів.

Важливою складовою захисту є регулярне резервне копіювання даних з обов'язковою перевіркою можливості їх відновлення у разі виникнення проблем. Сайт дотримується актуальних стандартів роботи з персональними даними, зокрема вимог GDPR, а також має прозору політику конфіденційності для своїх відвідувачів.

Таким чином, комплекс цих заходів гарантує безпеку, надійність та захист персональних даних користувачів веб-сайту ресторану «Маяк».

2.4 Адаптивність та UX-дизайн веб-сайту «Маяк»

Веб-сайт ресторану «Маяк» розроблений із урахуванням принципів адаптивного дизайну та сучасного UX. Це забезпечує зручний та приємний досвід для всіх категорій користувачів незалежно від типу пристрою, на якому переглядається сайт.

Інтерфейс автоматично підлаштовується під різні розміри екранів: сайт однаково коректно виглядає та функціонує як на комп'ютерах, так і на планшетах чи смартфонах, що відповідно зображено на рисунку 2.4. Ключові елементи (меню, кнопки, форми) оптимізовані для сенсорного керування: вони достатньо великі для натискання пальцем, мають зрозумілі підписи й чітку структуру.

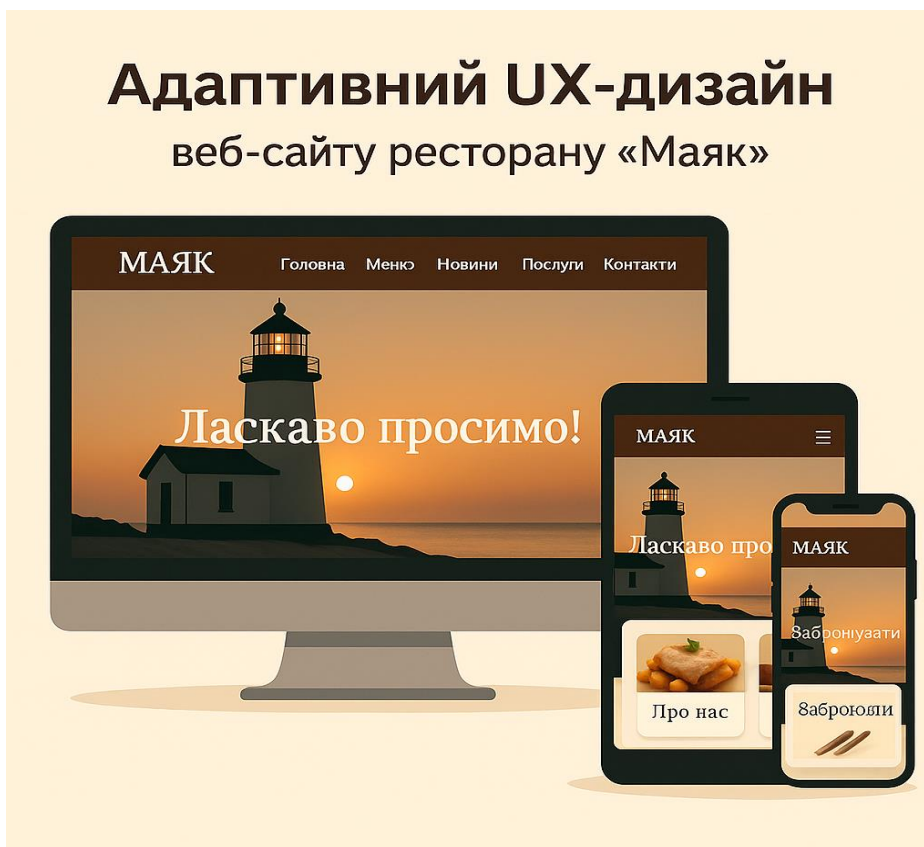


Рисунок 2.4 – Адаптивний UX-дизайн веб-сайту ресторану «Маяк»

Дизайн сайту мінімалістичний та візуально привабливий: використовується легка кольорова гама, якісні зображення, читабельна типографіка та достатньо «повітря» між елементами для комфортного сприйняття інформації. Всі інтерактивні елементи мають відгук на дії користувача — підсвічування кнопок, анімації повідомлень про успішні або помилкові операції. Це створює відчуття живого й динамічного інтерфейсу, підвищує задоволення від користування сайтом і зменшує ймовірність помилкових дій.

Особлива увага приділена доступності — сайт підтримує масштабування шрифтів, коректно працює зі скрінрідерами та дає змогу швидко перейти до основного контенту за допомогою клавіатурної навігації. Інтерфейс розроблено з урахуванням принципів інклюзивного дизайну: елементи керування мають зрозумілі підписи, достатній контраст кольорів і просту структуру сторінок, що забезпечує зручність використання для людей з різними потребами. Завдяки

цьому навіть користувачі із порушеннями зору чи моторики можуть повноцінно взаємодіяти з веб-ресурсом і замовляти послуги ресторану онлайн.

Завдяки впровадженню адаптивного та орієнтованого на користувача дизайну, сайт ресторану «Маяк» гарантує приємний і зрозумілий досвід для відвідувачів із будь-яких пристроїв [19]. Незалежно від того, чи користувач заходить із комп'ютера, планшета чи смартфона, усі функції й контент залишаються легкодоступними, а навігація — інтуїтивною. Сайт автоматично підлаштовується під розмір екрана, забезпечуючи збереження структури, пропорцій і зручності взаємодії.

Окрім цього, велика увага приділяється швидкості завантаження сторінок, оптимізації графіки та використанню сучасних технологій кешування. Завдяки цьому навіть при повільному інтернет-з'єднанні користувач отримує комфортний доступ до основних розділів і сервісів. Розміщення головних функцій у помітних місцях дозволяє швидко оформити замовлення, забронювати столик або дізнатися про нові акції.

Для забезпечення цілісного стилю всі елементи дизайну — від кнопок до заголовків і блоків меню — виконані у єдиній візуальній концепції, що формує впізнаваність бренду ресторану. Сучасна іконографіка, візуальні акценти та гармонійна кольорова палітра роблять сайт привабливим для всіх вікових категорій гостей.

Важливо й те, що дизайн був протестований на різних пристроях та у популярних браузерях, аби уникнути помилок відображення й гарантувати стабільну роботу сервісу. Таким чином, сайт ресторану «Маяк» не лише відповідає сучасним стандартам веб-розробки, а й є зразком комфортного, доступного й естетично привабливого онлайн-середовища для кожного відвідувача. Це сприяє підвищенню лояльності клієнтів, покращенню взаємодії з веб-ресурсом та зміцнює репутацію ресторану у цифровому просторі [20].

2.5 Управління замовленнями на веб-сайті ресторану «Маяк»

Система управління замовленнями на веб-сайті ресторану «Маяк» реалізована таким чином, щоб максимально спростити процес для користувачів та адміністрації закладу. Кожен відвідувач сайту може швидко сформувати нове замовлення, обравши необхідні страви із меню та заповнивши просту форму із контактними даними, як зображено на рисунку 2.5.

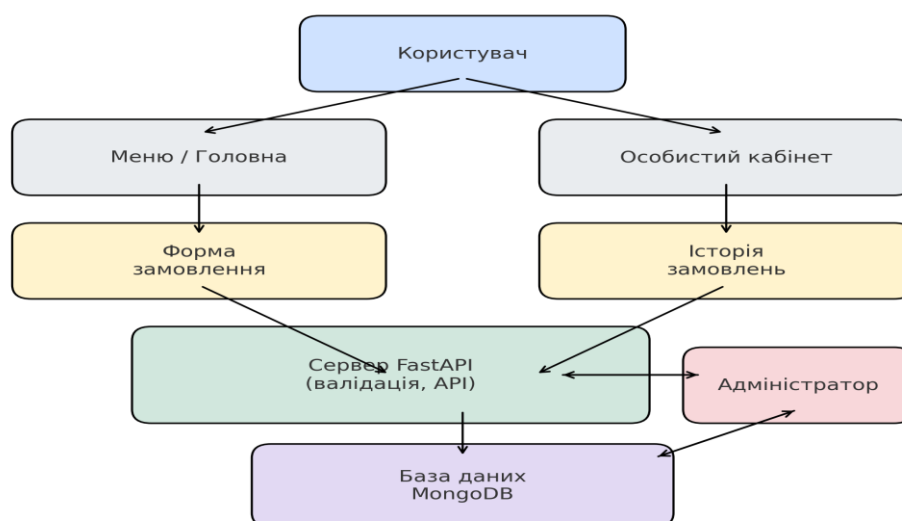


Рисунок 2.5 – Блок-схема «Управління замовленнями на веб-сайті ресторану «Маяк»»

Для забезпечення зручності та ефективності обробки замовлень на веб-сайті ресторану «Маяк» було обрано сучасний фреймворк FastAPI. Це один із найпотужніших та найшвидших інструментів для розробки веб-API на Python, який ідеально підходить для реалізації інтерактивних сервісів із високою продуктивністю.

Завдяки використанню FastAPI усі запити, пов'язані з оформленням, переглядом та оновленням замовлень, обробляються швидко та безпечно. FastAPI дозволяє легко створювати RESTful-ендпоінти для різних операцій: додавання нового замовлення, отримання списку замовлень, зміни їхнього

статусу або видалення. Усі ці процеси супроводжуються автоматичною валідацією даних та перевіркою прав доступу користувачів.

Важливою перевагою FastAPI є його інтеграція з асинхронними механізмами обробки запитів, що дає змогу обслуговувати велику кількість користувачів одночасно без втрати швидкодії. Крім того, FastAPI автоматично генерує інтерактивну документацію для всіх API-методів (Swagger UI), що значно полегшує підтримку та тестування сервісу.

Після оформлення замовлення вся інформація автоматично зберігається у захищеній базі даних. Користувач має змогу відстежувати статус свого замовлення у власному кабінеті, а за потреби — змінити чи скасувати заявку. Адміністратор ресторану отримує сповіщення про нові замовлення, переглядає їх у зручному інтерфейсі та оперативно реагує на запити клієнтів. Такий підхід дозволяє оптимізувати роботу персоналу, підвищити рівень обслуговування та забезпечити прозору комунікацію між рестораном і його гостями. Всі дані користувачів захищені сучасними механізмами безпеки, що гарантує конфіденційність і надійність сервісу.

Крім того, система розроблена таким чином, щоб бути масштабованою та готовою до інтеграції з іншими зовнішніми сервісами. Наприклад, у майбутньому можливе додавання модулів для онлайн-оплати через популярні платіжні системи, а також автоматичного надсилання повідомлень користувачам про статус замовлення через e-mail, SMS або месенджери. Адміністративна частина системи дозволяє працівникам ресторану не лише бачити перелік усіх замовлень, а й ефективно управляти ними за допомогою інтуїтивного інтерфейсу. Усі зміни у замовленнях фіксуються в логах, що дає змогу контролювати процес обробки та забезпечує прозорість для власників бізнесу.

Окрему увагу приділено адаптивності інтерфейсу — сайт і система управління замовленнями коректно працюють на різних пристроях: від комп'ютерів до смартфонів. Завдяки сучасним технологіям кешування даних і оптимізації запитів, сторінки завантажуються швидко навіть за нестабільного

інтернет-з'єднання, що особливо важливо для користувачів мобільних мереж. Під час розробки було впроваджено систему багаторівневої перевірки введених даних — це гарантує, що у замовленнях не буде помилок, а користувач отримає сповіщення у разі неправильного або неповного заповнення форми.

Ще однією важливою перевагою є зручність інтеграції з іншими модулями сайту. Наприклад, система дозволяє враховувати спеціальні акції, знижки для постійних клієнтів або промокоди. Усі ці функції легко додаються або змінюються через панель керування [22], не потребуючи втручання розробників у програмний код.

Таким чином, система управління замовленнями на веб-сайті ресторану «Маяк» — це сучасне, безпечне та масштабоване рішення, яке створює комфорт для гостей, полегшує роботу персоналу й дозволяє власникам ресторану тримати під контролем усі бізнес-процеси.

2.6 Висновок до другого розділу

У другому розділі кваліфікаційної роботи детально проаналізовано та обґрунтовано підходи до проектування основних елементів веб-сайту ресторану «Маяк», зосереджуючись на архітектурних рішеннях, розробці інтерфейсу, питаннях безпеки та організації процесу управління замовленнями. Проведене дослідження дало змогу сформувати комплексну, сучасну та гнучку архітектуру інформаційної системи, що базується на клієнт-серверній моделі та передбачає чіткий розподіл функціоналу між фронтенд і бекенд складовими.

В основі фронтенд-частини використано React — потужну JavaScript-бібліотеку, яка забезпечує створення швидкого, адаптивного та динамічного інтерфейсу користувача. Завдяки застосуванню компонентного підходу та сучасних принципів UI/UX, веб-сайт має логічну структуру, зрозумілу навігацію та привабливий вигляд, що значно підвищує рівень задоволеності відвідувачів і стимулює їх до подальших дій. Особливу увагу приділено питанням адаптивності інтерфейсу — сайт коректно працює на різних

пристроях та підтримує інклюзивний дизайн для користувачів із різними потребами.

Бекенд-система реалізована за допомогою FastAPI — сучасного Python-фреймворку, який забезпечує високу продуктивність, масштабованість та простоту інтеграції з іншими сервісами. Розділення обов'язків між фронтом і бекендом дозволяє реалізовувати гнучкі сценарії взаємодії, швидко оновлювати функціонал та підтримувати актуальність системи відповідно до нових вимог ринку.

Зберігання даних організовано на базі MongoDB, що забезпечує гнучкість роботи з напівструктурованими даними та високий рівень продуктивності при обробці великої кількості запитів. Такий підхід сприяє легкому масштабуванню системи та її адаптації до зростання кількості користувачів.

Важливою складовою проектування стала розробка комплексної системи безпеки. Впроваджено сучасні механізми автентифікації та авторизації на основі токенів JWT, які забезпечують захист особистих даних користувачів, надійний контроль доступу до ресурсів і захист від найпоширеніших кіберзагроз. Усі дані передаються через захищений протокол HTTPS, а паролі зберігаються у вигляді хешів з використанням стійких алгоритмів шифрування, що відповідає сучасним стандартам інформаційної безпеки.

Окрема увага у розділі приділена проектуванню інтерфейсу управління замовленнями. Завдяки впровадженню REST API, процеси створення, перегляду та редагування замовлень є максимально прозорими та зручними для користувачів. Автоматизація основних операцій дозволяє підвищити швидкість обробки заявок і зменшує кількість помилок у роботі персоналу. Інтерфейс адміністратора передбачає розширені можливості для моніторингу й аналізу поточних замовлень, що позитивно впливає на організацію бізнес-процесів у ресторані.

Підсумовуючи, можна стверджувати, що всі розглянуті рішення в другому розділі створюють надійну платформу для подальшого розвитку та вдосконалення веб-сайту ресторану «Маяк». Запропонована архітектура і

підходи до розробки забезпечують високий рівень зручності для користувачів, ефективність управління для адміністрації, а також відповідають сучасним тенденціям цифровізації та автоматизації сфери громадського харчування. Це сприятиме підвищенню конкурентоспроможності ресторану, розширенню його аудиторії та поліпшенню якості обслуговування гостей.

РОЗДІЛ 3. ТЕСТУВАННЯ ТА ВВЕДЕННЯ В ДІЮ В ВЕБ-САЙТУ «МАЯК»

3.1 Створена головна сторінка веб-сайту «Маяк»

Головна сторінка сайту ресторану «Маяк», як зображено на малюнку 3.1 спроектована так, щоб кожен відвідувач відчув комфорт та легкість взаємодії з ресурсом ще з перших секунд. В основі дизайну лежать сучасні принципи юзабіліті, увага до деталей і емоційна привабливість, що робить сайт не лише інформативним, а й дружнім до користувача.

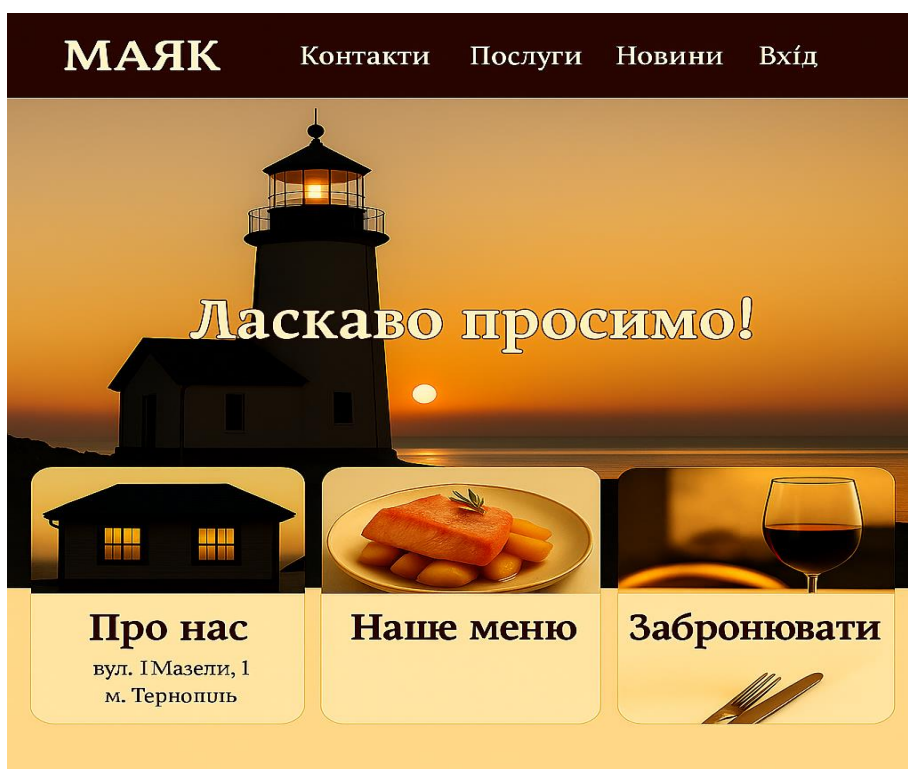


Рисунок 3.1 – Головна сторінка веб-сайту ресторану «Маяк»

У самому верхньому полі сайту знаходиться виразний логотип «Маяк», який слугує впізнаваною візитівкою закладу. Поруч із логотипом розташовано лаконічне головне меню: Контакти, Послуги, Новини, Вхід. Завдяки такій структурі користувач має миттєвий доступ до всіх ключових розділів. Контакти дає змогу швидко знайти адресу ресторану, телефон, e-mail та переглянути

розташування на карті. Послуги інформують про спектр додаткових можливостей — організацію свят, корпоративів, вечірок, кейтеринг. Новини — розділ із найсвіжішими подіями, акціями, анонсами та спецпропозиціями ресторану. Вхід — кнопка для переходу до сторінки авторизації, де відображається додаткове вікно, яке включає поле для введення логіна і пароля користувача, як подано приклад на рисунку 3.2, де відкривається додатковий функціонал для постійних клієнтів, дає змогу переглядати особистий кабінет, бонуси та історію бронювань.

The image shows a login form with a yellow background. It contains three input fields: the first is labeled 'Логін' (Login), the second is labeled 'Пароль' (Password), and the third is a dark blue button labeled 'Вхід' (Login/Enter).

Рисунок 3.2 – Форма для входу користувачів веб-сайту ресторану «Маяк»

Меню залишається завжди видимим і доступним, що дає змогу швидко переміщатися сайтом незалежно від того, де знаходиться користувач на сторінці.

Головний екран наповнений теплою атмосферою завдяки великому банеру із зображенням маяка на фоні заходу сонця над морем. Такий фон не лише підкреслює тематику закладу, а й викликає асоціації зі спокоєм, затишком та романтикою. Центральне місце займає привітальний напис «Ласкаво просимо!», який одразу створює доброзичливий настрій.

Під банером розміщені три великі інтерактивні блоки. Про нас містить зображення будівлі ресторану та коротку адресу. Перехід веде на сторінку з історією закладу, цінностями, традиціями, фото інтер'єру й атмосфери. Наше меню — апетитна світлина страви запрошує ознайомитися з розширеним асортиментом кухні. Меню містить докладний опис кожної позиції, ціни й

фото, усе зручно поділено на категорії. Забронювати оформлена у вигляді келиха та приборів — ця картка відкриває просту онлайн-форму бронювання столика: потрібно лише вибрати дату, час і кількість гостей, щоб здійснити резервування за кілька кліків.

В розділі «Про нас», розміщено інформаційний блок із контактами, відомостями про ресторан та можливостями співпраці як подано на рисунку 3.3.

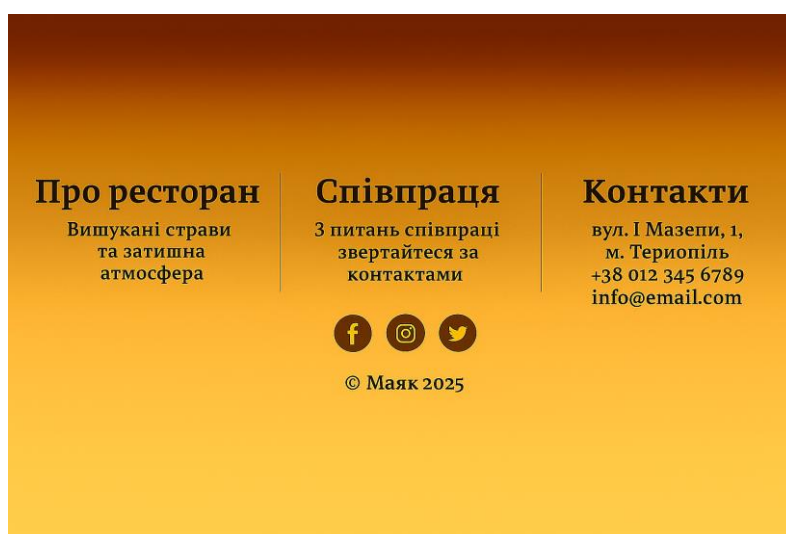


Рисунок 3.3 – Інформаційний блок веб-сайту ресторану «Маяк»

Інформаційний блок сайту ресторану «Маяк» організований так, щоб користувач міг швидко знайти ключову інформацію про заклад, дізнатись про можливості співпраці та легко зв'язатися з адміністрацією. Дизайн нижньої частини виконаний у теплих жовтих і коричневих відтінках, що перегукуються з основною палітрою сайту та створюють відчуття гармонії й завершеності [23]. Уся структура розподілена на три однакові колонки для максимальної зрозумілості та доступності.

Перша колонка присвячена ресторану. Тут у стислій формі відображено ідею та атмосферу закладу — це місце, де гості можуть насолодитися вишуканими стравами у затишній, гостинній обстановці. Такий опис одразу

задає настрій і підкреслює ключову перевагу «Маяк», яка полягає у поєднанні якісної кухні з комфортом.

Друга колонка - для партнерів та всіх, кого цікавить співпраця. Саме тут розміщене звернення до потенційних бізнес-партнерів, організаторів подій, постачальників або представників туристичних компаній. Усім їм пропонують з будь-яких питань співпраці звертатися за контактними даними, що демонструє відкритість ресторану до нових ідей і партнерських відносин.

У третій колонці зосереджені всі основні контактні дані для швидкого зв'язку з рестораном. Тут зазначено актуальну адресу у місті Тернопіль, зручний для дзвінків номер телефону та офіційна електронна скринька. Це дозволяє кожному гостю або партнеру миттєво знайти спосіб, як зв'язатися з адміністрацією, забронювати столик чи задати будь-яке питання.

Під цими основними секціями розміщено іконки соціальних мереж, які ведуть на офіційні сторінки ресторану у Facebook, Instagram та Twitter. Це не лише створює додаткові можливості для спілкування, а й дозволяє гостям слідкувати за новинами, акціями й подіями закладу [24]. Завершує підвал напис із копірайтом, який засвідчує оригінальність бренду та актуальність інформації.

Основна увага приділена розділу «Наше меню» — саме тут відвідувачі можуть зайти на окрему сторінку та ознайомитися з усіма основними стравами ресторану, як можна побачити на рисунку 3.4. Основна частина сторінки містить список категорій страв: піца, паста, рибні страви, м'ясні страви, салати, десерти, напої. Така структура дозволяє швидко зорієнтуватися у пропозиціях ресторану та обрати потрібний розділ. Окремим блоком виділено секцію «Популярне», де візуально представлені найпопулярніші позиції меню — фірмова піца «Маяк» та роли й суші «Філадельфія». Біля зображень популярних страв вказано їхні назви, а самі фото підібрані так, щоб виглядати апетитно та викликати гастрономічний інтерес у відвідувача. В правій частині розміщена акцентована кнопка «Акції», яка повідомляє про спеціальні пропозиції та знижки ресторану. Перейшовши до цього розділу, постійні відвідувачі можуть першими дізнаватися про вигідні сезонні пропозиції,

знижки на улюблені страви та напої, участь у програмах лояльності, накопичення бонусних балів і подарункові сертифікати для себе або друзів. Для зручності клієнтів у розділі акцій детально описані умови кожної пропозиції, вказані строки дії акцій та особливості участі для зареєстрованих користувачів.

Це дозволяє гостям завжди залишатися в курсі новинок, брати участь у тематичних заходах ресторану та отримувати персональні бонуси за часті замовлення чи святкування важливих подій саме у «Маяк». Постійні відвідувачі, які користуються особистим кабінетом, можуть підписатися на розсилку актуальних акцій і не пропускати жодної вигідної можливості. Таким чином, розділ «Акції» спрямований не лише на залучення нових клієнтів, а й на формування лояльної спільноти гостей, які цінують турботу та увагу з боку ресторану.



Рисунок 3.4 – Основна частина розділу «Наше меню» веб-сайту ресторану «Маяк»

Також на головній сторінці сайту розташований зручний розділ «Забронювати» із впізнаваною іконкою келиха та столових приборів.

Натиснувши на цю картку, гість автоматично переходить до онлайн-форми бронювання, як зображено на рисунку 3.5.

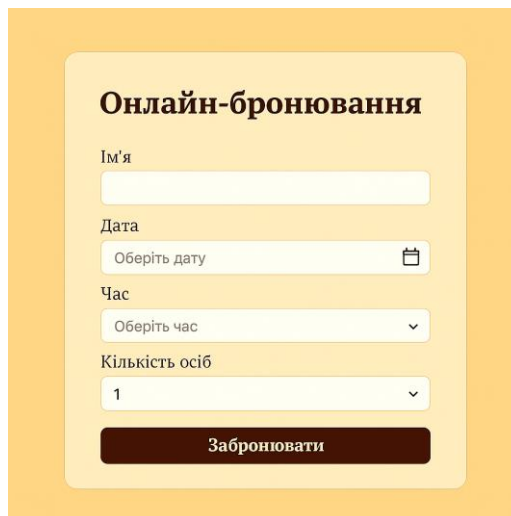
The image shows a mobile-friendly online reservation form titled "Онлайн-бронювання" (Online booking). The form is set against a light orange background. It contains several input fields: a text box for "Ім'я" (Name), a date picker for "Дата" (Date) with the placeholder "Оберіть дату" (Select date), a dropdown menu for "Час" (Time) with the placeholder "Оберіть час" (Select time), and another dropdown menu for "Кількість осіб" (Number of people) with the value "1" selected. At the bottom of the form is a dark brown button with the text "Забронювати" (Book).

Рисунок 3.5 – Онлайн-форма бронювання на веб-сайті ресторану «Маяк»

У цій формі потрібно вказати основні параметри майбутнього візиту: вибрати дату, час, кількість гостей, а за бажанням — залишити додатковий коментар чи особливі побажання (наприклад, розміщення біля вікна або потребу у дитячому стільчику). Форма інтуїтивна й адаптована під мобільні пристрої, тож заповнити її можна навіть «на ходу».

Після заповнення й підтвердження заявки система автоматично опрацьовує запит і надсилає підтвердження бронювання на вказаний номер телефону або e-mail. Адміністратор ресторану також отримує інформацію про нову заявку, тож у разі потреби може зв'язатися з гостем для уточнення деталей.

Завдяки онлайн-бронюванню користувачі заощаджують час, впевнені у наявності вільного столика та можуть планувати свій візит у зручний для себе спосіб. Це особливо зручно для організації сімейних свят, ділових зустрічей або романтичних вечорів у ресторані «Маяк».

Основна сторінка розроблена з урахуванням сучасних тенденцій вебдизайну, забезпечуючи користувачам зручність навігації та приємний

візуальний досвід. Всі елементи сторінки працюють разом, щоб створити позитивне враження про ресторан та заохотити відвідувачів до замовлення. Дизайн сайту гармонійно поєднує естетичність і функціональність: акцент на атмосферних зображеннях, якісна типографіка та інтуїтивно зрозуміла структура допомагають користувачеві легко орієнтуватися в розділах ресурсу. Верхня навігаційна панель із чітко виділеними розділами дозволяє швидко переходити до потрібної інформації, а інтерактивні блоки з меню, контактами і можливістю бронювання сприяють активній взаємодії із сайтом.

Особливу увагу приділено адаптивності сторінки — вона однаково добре відображається на комп'ютерах, планшетах і смартфонах, що забезпечує максимальне охоплення цільової аудиторії. Колірна гама та графічні елементи підкреслюють унікальний стиль ресторану й створюють у користувача відчуття затишку та гостинності ще до відвідування закладу. Інформативні розділи дозволяють ознайомитися з історією ресторану, переглянути меню з фотографіями страв, дізнатися про актуальні акції та новини, а також залишити заявку на бронювання столика або отримати консультацію. Усі ці компоненти сприяють підвищенню лояльності постійних гостей і приваблюють нових клієнтів, формуючи позитивний імідж ресторану «Маяк» у цифровому середовищі.

3.2 Реалізований функціонал особистого кабінету користувача на веб-сайті ресторану «Маяк»

Даний розділ присвячений опису та реалізації особистого кабінету користувача. Це один з основних елементів веб-сайту ресторану «Маяк», який дозволяє кожному відвідувачу після авторизації отримати доступ до персональних даних, історії замовлень, інформації про бонуси та знижки, а також швидко змінювати свої контактні дані.

Увійшовши у розділ «Особистий кабінет», як зображено на рисунку 3.6 користувач бачить чотири основні секції: «Профіль» — тут відображено

персональні дані (ім'я, номер телефону, e-mail), які можна редагувати; «Історія замовлень» — повний перелік усіх оформлених замовлень з датою, складом, статусом і сумою; «Бонуси і знижки» — інформація про накопичені бонуси, актуальні акції для постійних клієнтів і можливість використати бали при новому замовленні; «Налаштування» — розділ для зміни паролю та керування сповіщеннями. Користувач у будь-який момент може оновити контактні дані, змінити пароль або переглянути попередні замовлення, а система автоматично відображає доступні саме цьому гостю акції та спеціальні пропозиції.

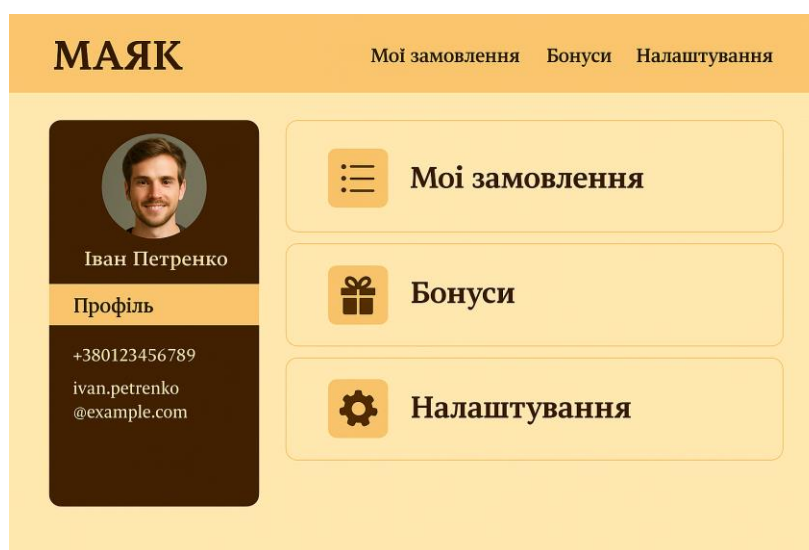


Рисунок 3.6 Головна сторінка особистого кабінету користувача на веб-сайті ресторану «Маяк»

У розділі «Профіль» користувач має можливість змінювати особисту інформацію, таку як ім'я, прізвище, електронна пошта та номер телефону, як зображено на рисунку рисунку 3.7. Даний функціонал є одним із ключових для забезпечення персоналізації облікового запису та подальшої коректної взаємодії з сервісом ресторану.

Інтерфейс редагування реалізовано у вигляді зручної та зрозумілої форми, де всі поля вже автоматично заповнені актуальними даними користувача. Це дозволяє оперативно вносити зміни у разі зміни контактної інформації, наприклад, при зміні номера телефону чи електронної пошти.

Перед збереженням змін усі введені дані проходять перевірку (валідацію) на коректність:

- електронна пошта повинна відповідати стандартному формату (наприклад, user@email.com);
- номер телефону — містити лише цифри та відповідати національному формату;
- поля імені та прізвища не можуть бути порожніми [25].

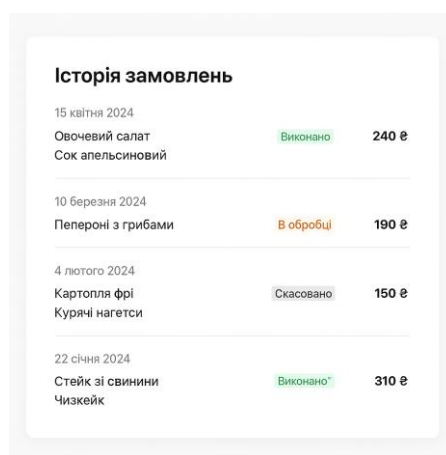
У випадку некоректного введення система інформує користувача про помилку та підсвічує проблемні поля. Завдяки цьому забезпечується мінімізація помилок при збереженні профілю та підвищується надійність переданої інформації.

Після успішного внесення змін нові дані автоматично зберігаються у базі даних, а користувач отримує повідомлення про успішне оновлення профілю. Крім того, оновлена інформація одразу використовується під час оформлення замовлень та для персоналізованих повідомлень від ресторану.

Рисунок 3.7 Редагування профіля користувача

Історія замовлень — це зведений журнал усіх транзакцій, які користувач оформив через сайт «Маяк». На рисунку 3.8 відображаються основні реквізити кожного замовлення: дата і точний час оформлення, повний склад (найменування страв, варіанти доповнень або модифікаторів), підсумкова сума з урахуванням бонусних списань чи акційних знижок, а також поточний статус.

Для зручності всі записи містять кольорові бейджі: «Виконано» підсвічено зеленим, «В обробці» — помаранчевим, «Скасовано» — сірим. Клікабельність рядка дає змогу розгорнути деталі й побачити розбивку за позиціями, спосіб оплати, адресу доставки чи номер зарезервованого столика. Передбачено фільтрацію за датою, сумою та статусом, а також пошук за ключовим словом, що дозволяє швидко знаходити потрібне замовлення серед великої історії. Існує опція «Повторити», яка копіює вміст у кошик, економлячи час постійним гостям; замовлення зі статусом «В обробці» можна скасувати або змінити адресу без дозволу адміністратору. У нижній секції система підраховує загальні витрати користувача за вибраний період і показує кількість накопичених бонусів, витрачених на кожне замовлення, що мотивує користувача планувати бюджет і частіше робити повторні покупки. Користувач може зберегти своє замовлення на пізніше, що дозволяє їм повернутися до оформлення замовлення у зручний для них час.



Історія замовлень		
15 квітня 2024	Овочевий салат Сок апельсиновий	Виконано 240 ₴
10 березня 2024	Пепероні з грибами	В обробці 190 ₴
4 лютого 2024	Картопля фрі Курячі нагетси	Скасовано 150 ₴
22 січня 2024	Стейк зі свинини Чизкейк	Виконано 310 ₴

Рисунок 3.8 Перегляд історії замовлень у особистому кабінеті

На фінальному етапі оформлення замовлення користувач має можливість вибрати зручний для себе спосіб оплати, а також підтвердити завершення замовлення. Приклад цього інтерфейсу показано на рисунку 3.10. Зокрема, клієнт може обрати оплату при отриманні замовлення. Додатково, на сторінці розміщено повідомлення про обробку персональних даних відповідно до чинної політики конфіденційності.

Щоб впевнитися, що замовлення успішно надіслано, було використано спеціально розроблене API, реалізоване на мові Python [26]. Виконуючи відповідний запит до серверу, можна отримати детальну інформацію про створене замовлення безпосередньо з бази даних та переконатися у його правильності та цілісності, як зображено на рисунку 3.9.

```
from fastapi import FastAPI

app = FastAPI()

def get_order('orders_id: int):
    # Отримати замовлення з бази даних
    return {"order_id": order_id,
            "name": "Мапрарита",
            "price": 230.0,
            "status": "Обробляється"}

app.post("/orders")
# Зберегти нове замовлення
return {"order_id": 1}

curl -X "GET" "http://127.0.0.1:8000/orders/1"

{"order_id": 1, "name": "Мапрарита", "price": 230.0, "status": "Об"
```

Рисунок 3.9 Отримане замовлення створене за допомогою форми на веб-сайті «Маяк»

Реалізація особистого кабінету користувача на веб-сайті ресторану «Маяк» забезпечує комплексний підхід до покращення взаємодії клієнта з онлайн-сервісом закладу. Завдяки сучасному інтерфейсу та зручній навігації користувачі можуть легко переглядати та змінювати свої особисті дані, зберігати декілька адрес доставки, додавати улюблені страви та переглядати історію попередніх замовлень.

Функціонал особистого кабінету сприяє підвищенню рівня персоналізації обслуговування, скорочує час оформлення замовлення, а також підвищує лояльність клієнтів до ресторану. Захист персональних даних та їх коректна обробка відповідає сучасним вимогам конфіденційності та стандартам безпеки.

3.3 Введення в експлуатацію веб-сайту «Маяк»

Введення в експлуатацію веб-сайту ресторану «Маяк» є важливим етапом у розвитку сучасного закладу громадського харчування, який прагне забезпечити своїм клієнтам максимальну зручність, швидкість та якість обслуговування. Запуск онлайн-платформи дає змогу автоматизувати процеси прийому, обробки й моніторингу замовлень, забезпечує прозору комунікацію між клієнтом і адміністрацією закладу, а також дозволяє значно розширити аудиторію за рахунок онлайн-продажів.

Веб-сайт створений з урахуванням сучасних вимог до функціональності, зручності користування та безпеки персональних даних. Користувачі отримують змогу переглядати меню, оформлювати замовлення, обирати спосіб оплати й доставки, відстежувати історію своїх покупок і накопичувати бонуси. Окремий акцент зроблено на реалізації особистого кабінету, який дозволяє зберігати персональні дані, декілька адрес доставки, улюблені страви та забезпечує швидкий доступ до історії замовлень.

Для впровадження веб-сайту було обрано сучасний технологічний стек: фронтенд - частина реалізована з використанням бібліотеки React, що дозволяє створювати швидкий та інтерактивний інтерфейс користувача, а бекенд - логіка побудована на FastAPI — високопродуктивному Python - фреймворку для створення REST API.

Фронтенд - частина веб-сайту реалізована з використанням сучасної бібліотеки React, що дає змогу забезпечити динамічний і зручний інтерфейс користувача, швидке реагування на дії клієнта та гнучкість у подальшій розробці функціоналу.

React забезпечує компонентний підхід, повторне використання логіки та швидкий клієнтський рендеринг.

У структурі фронтенду проєкту, побудованого з використанням бібліотеки React, виділяються такі основні каталоги та файли:

- `public/` — містить статичні файли, такі як `index.html`, `favicon`, маніфести PWA тощо. Усі ресурси з цієї папки копіюються у фінальну збірку без обробки.
- `src/` — основний вихідний код React-додатка.
- `components/` — багаторазові UI-компоненти (кнопки, модальні вікна, форми).
- `pages/` — окремі сторінки застосунку, такі як «Головна», «Меню», «Особистий кабінет» та інші.
- `services/` — обгортки для HTTP-запитів (наприклад, через бібліотеку `axios`) та робота з REST API.
- `assets/` — зображення, SVG-іконки, стилі SCSS або CSS Modules.
- `hooks/` — кастомні React-хуки (наприклад, `useAuth`, `useOrders`) для спрощення логіки управління станом.
- `utils/` — допоміжні функції, валідатори, константи, які використовуються у різних частинах застосунку.
- `App.js / App.jsx` — кореневий компонент додатка, відповідає за маршрутизацію між сторінками за допомогою бібліотеки `react-router-dom`.
- `index.js` — точка входу у застосунок, де відбувається рендер додатка у DOM-дерево браузера.
- `package.json` — файл із переліком залежностей проєкту, а також зі скриптами для збірки, запуску та тестування додатка (наприклад, `start`, `build`, `test`).
- `.env` — файл для зберігання змінних середовища, таких як URL до бекенд-API, токени, ключі тощо.

Така структура дозволяє організувати код у логічні та зрозумілі блоки, забезпечуючи масштабованість, повторне використання компонентів і зручність подальшої підтримки застосунку.

Бекенд - частина веб-сайту побудована з використанням FastAPI — сучасного високопродуктивного веб-фреймворку для Python, призначеного для створення RESTful API. Завдяки цьому фреймворку вдалося організувати

швидку та безпечну обробку запитів, інтегрувати взаємодію з базою даних, а також реалізувати авторизацію, аутентифікацію користувачів і захист персональної інформації. FastAPI також дозволяє легко масштабувати систему та забезпечує автоматичну генерацію документації для розробників, що спрощує підтримку й розширення платформи.

У кореневій папці бекенд-проєкту знаходяться такі ключові папки та файли:

- `app/` — Коренева папка бекенду, в якій розташовується основний код серверної частини.
- `main.py` — Точка входу в застосунок. Саме тут створюється об'єкт FastAPI, підключаються основні маршрути (роутери), `middleware` (проміжне ПЗ), налаштовується CORS для взаємодії з фронтом.
- `api/` — Підкаталог із роутерами, згрупованими за доменами бізнес-логіки: наприклад, `auth.py` (авторизація та аутентифікація), `orders.py` (робота із замовленнями), `menu.py` (управління меню ресторану).
- `api/v1/` — Папка для версійованих маршрутів REST API, що дозволяє у майбутньому підтримувати різні версії API одночасно.
- `models/` — Папка з ORM-моделями (SQLAlchemy або SQLModel), які відповідають структурам таблиць бази даних: `User`, `Dish`, `Order`, `OrderItem` тощо.
- `schemas/` — Містить Pydantic-схеми (наприклад, `UserCreate`, `OrderOut`), які використовуються для валідації та серіалізації вхідних та вихідних даних при роботі з API.
- `services/` — Тут знаходиться бізнес-логіка, сервіси для роботи з базою даних, а також інтеграція зі сторонніми сервісами (наприклад, надсилання електронних листів чи оплати через Stripe).
- `core/` — Каталог із налаштуваннями застосунку (`settings.py`), генерацією JWT-токенів, хешуванням паролів, конфігурацією CORS та іншими важливими службовими файлами.

- db/ — Містить налаштування підключення до бази даних PostgreSQL (або SQLite для розробки), сесійний менеджер, а також файли міграцій Alembic для оновлення структури БД.
- dependencies/ — Функції для DI (dependency injection) через Depends(). Тут реалізується перевірка прав доступу користувача, логіка пагінації, кешування запитів.
- tests/ — Каталог із pytest-тестами для модулів і сервісів (unit-тести, інтеграційні тести для API).
- requirements.txt або pyproject.toml — Файли зі списком залежностей Python-проєкту (FastAPI, Uvicorn, SQLAlchemy, Pydantic, pytest тощо).
- Dockerfile / docker-compose.yml — Офіційний Dockerfile для створення образу Python 3.11, налаштування гібриду Uvicorn-Gunicorn для продуктивного розгортання, а також файл docker-compose для запуску бекенду разом із Postgres, Nginx reverse-proxy тощо.

Запуск та подальша експлуатація веб-сайту ресторану «Маяк» не лише оптимізує внутрішні бізнес-процеси закладу, але й формує сучасний канал комунікації з клієнтами, підвищує рівень лояльності та конкурентоспроможності ресторану на ринку. Інтеграція гнучкої архітектури з сучасними технологіями дозволяє швидко впроваджувати нові сервіси та адаптуватися до потреб відвідувачів. Таким чином, впровадження веб-платформи створює надійну основу для цифрової трансформації ресторану та сталого розвитку в умовах сучасного ринку громадського харчування.

3.4 Висновок до третього розділу

У третьому розділі було комплексно розглянуто всі ключові етапи створення, тестування та введення у дію веб-сайту ресторану «Маяк». На початковому етапі було описано розробку головної сторінки, що виконана з урахуванням сучасних принципів UX/UI дизайну, адаптивності та

інформативності, що забезпечує легкий доступ користувачів до основної інформації про ресторан, меню та контактні дані.

Наступним важливим етапом стала реалізація особистого кабінету користувача, який став основним інструментом для персоналізації взаємодії між клієнтом і закладом. В особистому кабінеті реалізовано можливість редагування персональних даних, зберігання декількох адрес доставки, перегляду історії замовлень, а також формування списку улюблених страв. Завдяки цьому функціоналу користувачі отримали зручний і безпечний спосіб керувати своїми замовленнями, що значно підвищує лояльність клієнтів та сприяє повторним покупкам.

Окрему увагу було приділено технічному впровадженню веб-платформи. Детально розглянуто архітектуру фронтенд- та бекенд-частин проєкту. Фронтенд побудовано з використанням сучасної бібліотеки React, що забезпечує швидку взаємодію, масштабованість і легкість підтримки. Бекенд-логіка реалізована на базі FastAPI, що гарантує високу продуктивність, гнучкість у розробці API, ефективну роботу з базами даних і безпечну обробку персональних даних. Проведено тестування всіх основних функціональних модулів, перевірено коректність взаємодії між клієнтською та серверною частинами, а також протестовано типові сценарії роботи користувача.

Введення веб-сайту в експлуатацію дозволило оптимізувати бізнес-процеси ресторану, автоматизувати прийом і обробку замовлень, забезпечити високий рівень сервісу та захисту даних. Веб-платформа стала надійним інструментом для комунікації з клієнтами, маркетингу та просування послуг ресторану в онлайн-середовищі. Сайт відкрив нові можливості для розвитку ресторану «Маяк», розширення цільової аудиторії та підвищення конкурентоспроможності закладу на ринку.

Загалом, результати, отримані в межах третього розділу, підтверджують доцільність цифрової трансформації закладів громадського харчування за допомогою сучасних веб-технологій та створення власних онлайн-платформ для підвищення якості обслуговування клієнтів і ефективності бізнесу.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Працездатність людини – оператора

У процесі роботи людина переживає різні функціональні стани, які зумовлюють різні рівні її працездатності.

Під працездатністю людини розуміють можливість її виконувати певну роботу з необхідною якістю та у встановлений час. Працездатність людини залежить як від зовнішніх факторів, так і від внутрішнього стану (внутрішні фактори).

До зовнішніх факторів належать: кількість та форма отриманої інформації, зручність робочого місця, характер взаємовідносин в колективі, вплив факторів середовища існування.

До внутрішніх факторів належать: рівень підготовки, тренованість людини та її емоційна стійкість.

Розглядаючи зміни функціонального стану та якості роботи людини у процесі одного трудового циклу (зміни), виділяють 4 фази працездатності: пристосування до праці, стійкої працездатності, субкомпенсації, втоми. Тривалість усіх фаз та усього циклу роботи залежить від рівня підготовки людини до роботи на рисунку 4.1.

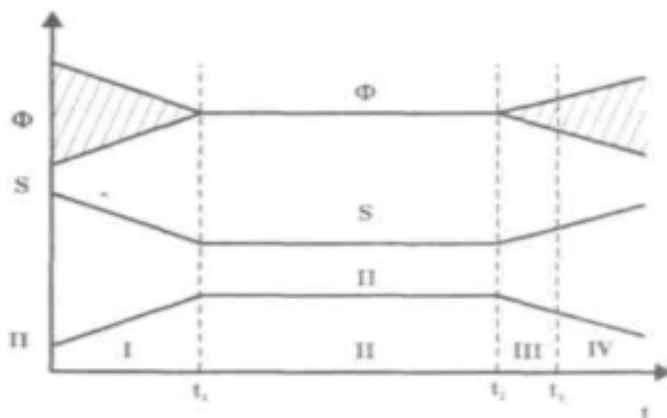


Рисунок 4.1 – Фази працездатності

Фаза пристосування до праці ($0 - t_1$) - це час, протягом якого людина адаптується до умов праці. Основний показник ефективності праці поступово досягає свого встановленого значення. Тривалість періоду пристосування організму до умов праці залежить від багатьох факторів, серед яких основними є інтенсивність роботи (чим інтенсивніша робота, тим цей період коротший) та рівень готовності людини до майбутньої роботи.

Фаза стійкої працездатності ($t_1 - t_2$) характеризується найвищою якістю праці при оптимальних рівнях функціонування фізіологічних систем організму. Тривалість цього періоду залежить від інтенсивності роботи. Чим інтенсивніша праця, тим коротший цей період. Найоптимальніша динамічна робота, коли цей період може бути в десятки разів довшим, ніж при статичній діяльності.

На процес стійкої працездатності впливають емоції. Негативні (страх, невпевненість, поганий настрій) знижують працездатність. Позитивні (впевненість, спокій, бадьорий настрій) значно продовжують період стійкої працездатності.

Продовження періоду стійкої працездатності можна забезпечити:

- оптимальним рівнем напруги психофізіологічних функцій;
- комфортними умовами праці;
- правильним поєднанням режимів праці та відпочинку;
- емоційним розвантаженням;
- використанням тонізуючих напоїв (кава, чай), фармакологічних засобів, зокрема препаратів рослинного походження (вітаміни, препарати, які впливають на енергетичні та метаболічні процеси);
- інформуванням людини про наслідки її діяльності, наглядом та контролем її роботи.

Фаза субкомпенсації ($t_2 - t_3$) розглядається як початок розвитку втоми. В цей період якість праці ще зберігається на високому рівні, але тільки за рахунок перенапруги відповідних функцій організму.

Фаза втоми (з моменту t_3) характеризується чітко вираженим зниженням якості роботи при подальшому погіршенні функціонального стану людини.

Об'єктивними показниками втоми є зміна частоти пульсу, дихання, зорової та слухової чутливості.

Наступною фазою життєдіяльності людини повинна бути фаза відновлення працездатності (відпочинку), яка може тривати від декількох хвилин до декількох годин і навіть декілька діб [33].

4.2 Затрати на заходи щодо покращення умов та охорони праці.

Витрати на заходи щодо поліпшення умов і охорону праці можна поділити на:

- витрати, пов'язані з відшкодуванням потерпілим втрат внаслідок травм і професійних захворювань;
- витрати на компенсацію за роботу в несприятливих умовах, що не відповідають санітарним нормам (пільги за важкі і шкідливі умови);
- витрати на попередження та профілактику травматизму і професійних захворювань;
- витрати на ліквідацію наслідків аварій та нещасних випадків;
- витрати на штрафи та інші відшкодування. Складові витрат подані в табл. 4.1.

Розглядаючи витрати на охорону праці, слід зазначити, що їх можна поділити на:

- доцільні, спрямовані на збереження здоров'я працівників, на раціональне витрачання ними життєвих сил під час роботи та на відновлення працездатності;
- частково доцільні, які включають витрати на пільги і компенсації за несприятливі умови праці;
- недоцільні витрати, що обумовлюють підвищення собівартості продукції, зниження її обсягу тощо.

Доцільні витрати забезпечують поліпшення умов праці і зростання її продуктивності, частково доцільні і недоцільні витрати призводять до збитків підприємства та зниження ефективності виробництва.

Дослідження свідчать: на підприємствах значно більше витрат припадає на пільги та компенсації, що пов'язані з небезпечними і шкідливими умовами праці, ніж на техніку безпеки, заходи щодо запобігання виробничому травматизму і захворюваності та нормалізацію умов праці. Співвідношення між витратами на поліпшення умов і охорону праці і витратами на доплати за несприятливі умови праці, пільгові пенсії та додаткові відпустки становить 1:10, а іноді й більше.

Таблиця 4.1 – Складові витрат

№ груп витрат	Назва витрат	Складові витрат
I	Відшкодування потерпілим внаслідок травм і професійних захворювань	Тимчасова непрацездатність; одноразова допомога (включаючи членів сімей і утриманців загиблих); моральна шкода; відшкодування витрат лікувальним закладам; санаторно-курортне обслуговування; протезування, придбання транспортних засобів, витрати на соціальну допомогу інвалідам; доплати до попереднього заробітку в разі переведення на легшу роботу; пенсії інвалідам і утриманцям загиблих.
II	Пільги та компенсації за працю у важких і шкідливих умовах	Додаткові відпустки; скорочений робочий день; лікувально-профілактичне харчування; безкоштовна видача молока чи інших рівноцінних продуктів; підвищені тарифні ставки; пільгові пенсії.

Продовження таблиці 1.1

III	Витрати на профілактику травматизму та професійних захворювань	Витрати на заходи з охорони праці за рахунок джерел фінансування, регламентованих нормативними актами держави; витрати на заходи з охорони праці за рахунок підприємства.
IV	Витрати на ліквідацію наслідків аварій та нещасних випадків на виробництві	Відшкодування вартості зіпсованого устаткування, інструментів, зруйнованих будівель, споруд; рятування потерпілих; виплати заробітної плати і доплати за час простою; вартість ремонту зіпсованого обладнання, машин і механізмів, будівель і споруд; вартість підготовки та перепідготовки працівників замість вибулих внаслідок загибелі чи інвалідності.
V	Штрафи та інші відшкодування	Штрафи за наявність нещасних випадків та приховування від обліку потерпілих; штрафи за недотримання нормативних вимог щодо безпеки праці; штрафи на працівника за порушення вимог законодавства та нормативних актів з охорони праці; компенсації за час вимушеного простою через небезпеку виконання робіт та через припинення робіт органами державного нагляду за охороною праці; штрафи, пені, виплати за недотримання договірних зобов'язань перед іншими підприємствами, установами і організаціями; тощо.

В умовах недосконалості ринкових механізмів усі ці витрати відносять на собівартість продукції, і, в результаті, за недбале ставлення до охорони праці на підприємствах розплачуються не їх керівники, а суспільство. Діюча система пільг і компенсацій не спонукає керівників поліпшувати умови праці, тому що ці витрати розкладаються на всіх споживачів і не впливають на економічні результати роботи підприємства

З іншого боку, штрафні санкції, що сплачуються підприємством у разі незадовільної роботи з охорони праці та при наявності фактів травмування і профзахворювань працівників, нині досить значні, а тому змушують будь-якого роботодавця (уповноважений ним орган) серйозно замислитися, що краще – зазнавати збитків (які часом можуть призвести навіть до банкрутства), не займаючись охороною праці, чи, не конфліктуючи з законом, своєчасно вкладати кошти у профілактичні заходи, що зберігають життя і здоров'я людей.

Звичайно, доцільно обрати другий варіант. Адже перелік штрафних санкцій та інших економічних втрат підприємства передбачає: штрафи, що накладаються на підприємство органами державного нагляду за охороною праці; штрафи за кожен нещасний випадок на виробництві або професійне захворювання; збільшені страхові виплати, що повинні здійснюватися роботодавцем у випадку незадовільної роботи з охорони праці у разі зростання виробничого травматизму та професійної захворюваності; витрати на виконання рятувальних робіт під час аварій та нещасних випадків, проведення розслідування та експертизи їх причин, поховання загиблих, на складання санітарно-гігієнічної характеристики робочого місця працівника, який одержав професійне захворювання, тощо.

Значними можуть бути витрати на пільги й компенсації, передбачені чинним законодавством і колективними договорами: за важкі та шкідливі умови праці (додаткові відпустки, лікувально-профілактичне харчування, молоко чи рівноцінні йому харчові продукти); оплата регламентованих перерв санаторно-оздоровчого призначення, що надаються під час виконання вібронебезпечних та інших шкідливих робіт, тощо. Отже, ці витрати також

повинні враховуватися роботодавцем у загальній сумі матеріальних втрат, що можуть мати місце на підприємстві через недостатню увагу до розв'язання проблем охорони праці.

4.3 Висновок до четвертого розділу

У четвертому розділі кваліфікаційної роботи було комплексно розглянуто питання безпеки життєдіяльності та охорони праці у процесі виконання сучасних цифрових завдань. Особливу увагу приділено збереженню працездатності оператора (розробника), ергономічній організації робочого місця, дотриманню режиму праці та відпочинку, а також впливу мікроклімату і психоемоційного навантаження на здоров'я та ефективність працівника.

На практичному прикладі описано наслідки недотримання гігієнічних та санітарних вимог, що призвели до погіршення самопочуття та зниження продуктивності. Застосування коригувальних заходів — оновлення робочого місця, впровадження режиму активного відпочинку та фізичної активності — позитивно вплинуло на відновлення працездатності і покращення результатів роботи.

Також у розділі було розглянуто витрати на заходи щодо покращення умов праці та охорони праці, їх класифікацію за ефективністю, а також показники економічної та соціальної доцільності. Визначено, що інвестування у безпечне робоче середовище дозволяє знизити рівень плинності кадрів, зменшити кількість захворювань та нещасних випадків, а також підвищити загальну ефективність роботи підприємства.

Таким чином, безпека життєдіяльності та охорона праці виступають ключовими складовими не лише збереження здоров'я працівників, а й забезпечення стабільного, якісного і результативного виконання професійних завдань.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було повністю досягнуто поставлену мету — здійснено всебічний аналіз предметної області ресторанного бізнесу, визначено актуальні потреби сучасного ринку та сформовано вичерпний перелік вимог до веб-сайту ресторану «Маяк». Протягом дослідження було розглянуто ключові аспекти цифрової трансформації закладів громадського харчування, з урахуванням змін у поведінці споживачів, зростання значення онлайн-сервісів і необхідності створення конкурентоспроможного веб-ресурсу.

У роботі було вирішено такі основні завдання:

- Проведено детальний аналіз існуючих аналогів сайтів ресторанного бізнесу, визначено їхні сильні та слабкі сторони, а також сучасні тенденції у сфері UX/UI-дизайну та функціональності;
- Обґрунтовано доцільність створення власного сайту ресторану «Маяк» як ключового інструменту просування бренду, залучення нових клієнтів і підвищення якості сервісу;
- Сформовано комплекс функціональних і нефункціональних вимог до майбутнього веб-ресурсу, включаючи інтеграцію з платіжними системами, адаптивність, захист даних користувачів та підтримку адміністративного керування контентом;
- Визначено оптимальний технологічний стек для реалізації веб-платформи — сучасні фронтенд та бекенд інструменти, реляційні й документоорієнтовані бази даних, хмарні сервіси для розгортання та підтримки ресурсу.

Досягнуті результати свідчать про ефективність комплексного підходу до розробки інформаційних систем для закладів громадського харчування. Представлений проект веб-сайту ресторану «Маяк» має прикладну цінність: забезпечує зручний канал комунікації з клієнтами, автоматизує ключові бізнес-процеси та формує сучасний імідж закладу у цифровому середовищі.

В цілому, виконана робота підтвердила актуальність впровадження цифрових технологій для підвищення конкурентоспроможності та розвитку ресторанного бізнесу. Вона демонструє сформовані навички аналізу, системного проектування, сучасної веб-розробки та розуміння вимог цільової аудиторії. Реалізований підхід і запропоновані рішення можуть бути використані для подальшої модернізації закладів громадського харчування та впровадження інновацій у сфері сервісу.

ПЕРЕЛІК ДЖЕРЕЛ

- 1 Kunanets, N., Dobrovolska, V., Filippova, N., Parviz, K., Lypak, H., Duda, O., ... & Dubrovina, L. (2020, September). Designing the Repository of Documentary Cultural Heritage. In Conference on Computer Science and Information Technologies (pp. 1034-1044). Cham: Springer International Publishing. Doi: 10.1007/978-3-030-63270-0_70 (дата звернення: 19.06.2025).
- 2 17 прикладів найкращих сайтів ресторанів та кафе. URL: <https://impulse-design.com.ua/ua/17-primerov-luchshikh-sajtov-restoranol-i-kafe.html> (дата звернення: 19.06.2025).
- 3 Адаптивний веб-дизайн URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/CSS_layout/Responsive_Design (дата звернення: 19.06.2025).
- 4 Duda, O., Pasichnyk, V., Lypak, H., ...Matsiuk, O., Mudrokha, V. Formation of integrated repositories of social and communication data by consolidating the resources of museums, libraries and archives in smart cities projects. CEUR Workshop Proceedings, 2021, 2870, pp. 1420–1430. URL: <http://ceur-ws.org/Vol-2870/paper104.pdf> (дата звернення: 19.06.2025).
- 5 Lypak, H., Kunanets, N., Pasichnyk, V., Veretennikova, N. Digitization Project for Historical and Cultural Heritage. 2020 IEEE 15th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2020 - Proceedings, 2020, 2, pp. 194–198, URL: <https://ieeexplore.ieee.org/document/9321993> (дата звернення: 19.06.2025).
- 6 Teslyuk, T., Teslyuk, V., Lypak, H., Kunanets, N., Veretennikova, N. A mobile museum guide application. CEUR Workshop Proceedings, 2020, 2631, pp. 314–326. <http://ceur-ws.org/Vol-2631/paper24.pdf> ISSN 1613-0073.
- 7 Актори та діаграми прецедентів системи консолідації соціокомунікаційних інформаційних ресурсів «Розумних міст» / Пасічник В. В., Кунанець Н. Е., Дуда О. М., Липак Г. І., О Мацюк. В., Семенюк В. В. //

Науковий вісник НЛТУ України: зб наук-техн. праць. – 2017. – Вип. 27(10). – С. 129–136. (дата звернення: 19.06.2025).

8 JSON Web Tokens для захищених API. URL: <https://auth0.com/docs/secure/tokens/json-web-tokens> (дата звернення: 19.06.2025).

9 Introduction to client-side frameworks URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/CSS_layout/Responsive_Design (дата звернення: 19.06.2025)

10 Вибір бекенд-технологій для розробників. URL: <https://www.geeksforgeeks.org/how-to-choose-the-right-backend-technology/> (дата звернення: 19.06.2025).

11 Огляд можливостей FastAPI для сучасних API. URL: <https://medium.com/codenx/fastapi-really-fast-and-modern-7a1796597b60> (дата звернення: 19.06.2025).

12 What is a relational database? URL: <https://www.ibm.com/think/topics/relational-databases> (дата звернення: 19.06.2025).

13 Розробка ефективного веб-сайту ресторану: поради та найкращі практики URL: <https://restaurantify.com/designing-an-effective-restaurant-website-tips-and-best-practices/> (дата звернення: 19.06.2025).

14 Валідація форм у React: react-hook-form URL: <https://www.freecodecamp.org/news/how-to-create-forms-in-react-using-react-hook-form/> (дата звернення: 19.06.2025).

15 Структура проєкту React для початківців URL: <https://dou.ua/forums/topic/44659/> (дата звернення: 19.06.2025).

16 Дизайн інтерфейсу користувача (UI). URL: <https://www.interaction-design.org/literature/topics/ui-design> (дата звернення: 19.06.2025).

- 17 REST API: найкращі практики проєктування URL: <https://www.freecodecamp.org/news/rest-api-best-practices-rest-endpoint-design-examples/> (дата звернення: 19.06.2025).
- 18 Збої реєстрації та моніторингу безпеки URL: https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/ (дата звернення: 19.06.2025).
- 19 Як створити сайт для онлайн-замовлень їжі. URL: <https://www.cleveroad.com/blog/how-to-make-a-food-ordering-website/> (дата звернення: 19.06.2025).
- 20 Адаптивний веб-дизайн (RWD) та користувацький досвід. URL: <https://www.nngroup.com/articles/responsive-web-design-definition/> (дата звернення: 19.06.2025).
- 21 Компоненти інтерфейсу користувача для швидшого створення потужних адміністративних панелей. URL: <https://coreui.io/> (дата звернення: 19.06.2025).
- 22 Посібник з архітектури програмного забезпечення. URL: <https://martinfowler.com/architecture/> (дата звернення: 19.06.2025).
- 23 10 евристик юзабіліті для дизайну користувацького інтерфейсу. URL: <https://www.nngroup.com/articles/ten-usability-heuristics/> (дата звернення: 19.06.2025).
- 24 Повний посібник із соціальних мереж для ресторанів та барів. URL: <https://sproutsocial.com/insights/bars-restaurants-social-media-guide/> (дата звернення: 19.06.2025).
- 25 Перевірка форми на стороні клієнта. URL: https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Forms/Form_validation (дата звернення: 19.06.2025).
- 26 Використання FastAPI для створення веб-API на Python. URL: <https://realpython.com/fastapi-python-web-apis/> (дата звернення: 19.06.2025).

- 27 Шпаргалка з UX-дослідження. URL: <https://www.nngroup.com/articles/ux-research-cheat-sheet/> (дата звернення: 19.06.2025).
- 28 Інтеграція платіжних систем у ресторанных сайтах. URL: <https://www.uptech.team/blog/how-to-create-online-payment-website> (дата звернення: 19.06.2025).
- 29 Мистецтво розробки веб-сайту ресторану [поради, приклади, шаблони]. URL: https://www.slammedialab.com/post/designing-restaurant-website?utm_source=chatgpt.com (дата звернення: 19.06.2025).
- 30 How to Tackle Your Restaurant Website Design Like a Pro. URL: <https://daily.dev/blog/3d-modeling-basics-for-developers> (дата звернення: 19.06.2025).
- 31 New Data: 6 Ways Guests Judge Your Restaurant's Website (And What to Do About It). URL: https://www.owner.com/blog/restaurant-website-design?utm_source=chatgpt.com (дата звернення: 19.06.2025).
- 32 Best restaurant website designs: 19 unique examples. URL: <https://webflow.com/blog/restaurant-website-design> (дата звернення: 19.06.2025).
- 33 Працевдатність людини-оператора. URL: <https://studfile.net/preview/4474247/page:11/> (дата звернення: 19.06.2025).
- 34 Витрати на покращення умов і охорону праці. URL: <https://studies.in.ua/bjd-gandzyuk/913-53-vitrati-na-pokraschennya-umov-ohoronu-prac.html>.
- 35 Навчально-методичний посібник. URL: https://elartu.tntu.edu.ua/bitstream/lib/48496/1/Navchal%ca%b9no_metodychnyy_%20posibnyk_%202025.pdf (дата звернення: 19.06.2025).

ДОДАТКИ

Програмний код для стартової сторінки веб-сайту ресторану «Маяк»

```

import React from 'react';
import { BrowserRouter as Router, Route, Routes } from 'react-router-dom';

import Header from './components/Header';
import Footer from './components/Footer';

import HomePage from './components/HomePage';
import MenuPage from './components/MenuPage';
import AboutPage from './components/AboutPage';
import ContactsPage from './components/ContactsPage';
import ReservationPage from './components/ReservationPage';
import CartPage from './components/CartPage';
import ProfilePage from './components/ProfilePage';
import Login from './components/Login';
import Register from './components/Register';

import AuthProvider from './contexts/AuthProvider';
import CartProvider from './contexts/CartProvider';

function App() {
  return (
    <AuthProvider>
      <CartProvider>
        <Router>
          <div className=«App»>
            <Header />
            <div className=«content»>
              <Routes>
                <Route path=«/» element={<HomePage />} />
                <Route path=«/menu» element={<MenuPage />} />
                <Route path=«/about» element={<AboutPage />} />
                <Route path=«/contacts» element={<ContactsPage />} />
                <Route path=«/reservation» element={<ReservationPage />} />
                <Route path=«/cart» element={<CartPage />} />
                <Route path=«/profile» element={<ProfilePage />} />
                <Route path=«/login» element={<Login />} />
                <Route path=«/register» element={<Register />} />
              </Routes>
            </div>
            <Footer />
          </div>
        </Router>
      </CartProvider>
    </AuthProvider>
  );}export default App;

```

**Програмний код що реалізує механізм автентифікації користувачів
на веб-сайті ресторану «Маяк»**

```
import { useContext } from 'react';
import { AuthContext } from '../contexts/AuthProvider';

const Profile = () => {
  const { token, logout, isAuthenticated } =
    useContext(AuthContext);

  return isAuthenticated ? (
    <div>
      <h2>Ви увійшли в систему</h2>
      <button onClick={logout}>Вийти</button>
    </div>
  ) : (
    <p>Увійдіть, щоб переглянути профіль</p>
  );
};
```

**Програмний код головної сторінки особистого кабінету
користувача на веб-сайті ресторану «Маяк»**

```
import { useContext, useEffect, useState } from 'react';
import { AuthContext } from '../contexts/AuthProvider';

const Dashboard = () => {
  const { token, logout, isAuthenticated } =
    useContext(AuthContext);

  /* локальний стан для даних */
  const [profile, setProfile] = useState(null);
  const [reservations, setReservations] = useState([]);

  /* бекенд-URL; змініть за потреби */
  const API = 'http://localhost:4000/api';

  /* хелпер із токеном */
  const api = (path, opts = {}) => {
    opts.headers = { ...opts.headers, Authorization: `Bearer
    ${token}` };

    return fetch(API + path, opts).then(r => r.ok ? r.json() :
    Promise.reject());
  };

  /* завантажити дані після монтування */
  useEffect(() => {
    if (!isAuthenticated) return;
    (async () => {
      try {
        const prof = await api('/profile');
        const resv = await api('/reservations');
        setProfile(prof);
        setReservations(resv);
      }
    })();
  });
}
```

```

        } catch { logout(); }
    ))();
}, [isAuthenticated]);

/* _____ render _____ */
if (!isAuthenticated) {
    return <p>Увійдіть, щоб переглянути кабінет</p>;
}

if (!profile) {
    return <p>Завантаження...</p>;
}

return (
    <div style={{ maxWidth: 600, margin: '2rem auto', padding: '0
1rem' }}>
        <h2>Вітаємо, {profile.name}!</h2>

        {/* профіль */}
        <section style={card}>
            <h3>Мій профіль</h3>
            <p><strong>Email:</strong> {profile.email}</p>
            <p><strong>Телефон:</strong> {profile.phone}</p>
        </section>

        {/* бронювання */}
        <section style={card}>
            <h3>Мої бронювання</h3>
            {reservations.length === 0 ? (
                <p>Бронювань поки немає.</p>
            ) : (
                <ul>
                    {reservations.map(r => (
                        <li key={r.id}>

```



```

         <strong>{r.date}</strong> o {r.time} —
        {r.guests} гостей (<em>{r.note}</em>)
      </li>
    )})
  </ul>
})
</section>

  <button onClick={logout}>Вийти</button>
</div>
);
};

/* базові стилі «картки» */
const card = {
  background: '#fff',
  padding: '1rem 1.5rem',
  borderRadius: '1rem',
  boxShadow: '0 4px 16px rgba(0,0,0,.1)',
  marginBottom: '1.5rem'
};

export default Dashboard;

```

Програмний код редагування профіля користувача на веб-сайті ресторану «Маяк»

```
import EditProfile from './EditProfile';

const [editMode, setEditMode] = useState(false);

{editMode
  ? <EditProfile onSave={u => {
      setProfile(u);          // оновлюємо дані на сторінці
      setEditMode(false);
    }} />
  : <>
    {/* поточний профіль */}
    <p><strong>Ім'я:</strong> {profile.name}</p>
    <p><strong>Email:</strong> {profile.email}</p>
    <p><strong>Телефон:</strong> {profile.phone}</p>
    <button                      onClick={ ()                      =>
setEditMode(true) }>Редагувати</button>
    </>
  }
}
```

Програмний код оформлення замовлення на веб-сайті ресторану «Маяк»

```

import { createContext, useEffect, useState } from 'react';

/* ----- створюємо контекст із початковими значеннями -----
--- */
export const AuthContext = createContext({
  token          : null,
  login          : () => {},
  logout         : () => {},
  isAuthenticated: false
});

const AuthProvider = ({ children }) => {
  /* зчитуємо токен з localStorage тільки один раз (lazy-init) */
  const [token, setToken] = useState(() =>
    localStorage.getItem('token'));

  /* ----- дії ----- */
  const login = newToken => {
    localStorage.setItem('token', newToken);
    setToken(newToken);
  };

  const logout = () => {
    localStorage.removeItem('token');
    setToken(null);
  };

  /* ----- синхронізація між вкладками ----- */
  useEffect(() => {
    const sync = e => {
      if (e.key === 'token') setToken(e.newValue);
    };
  });

```

```
    window.addEventListener('storage', sync);
    return () => window.removeEventListener('storage', sync);
  }, []);

const isAuthenticated = Boolean(token);

return (
  <AuthContext.Provider
    value={{ token, login, logout, isAuthenticated }}>
    {children}
  </AuthContext.Provider>
);
};

export default AuthProvider;
```

Програмний код історії замовлень у особистому кабінеті

```
import { useContext, useEffect, useState } from 'react';
import { AuthContext } from '../contexts/AuthProvider';

export default function OrderHistory() {
  const { token, isAuthenticated, logout } =
    useContext(AuthContext);

  const [orders, setOrders] = useState(null);
  const API = 'http://localhost:4000/api';

  /* helper - fetch is JWT */
  const api = path =>
    fetch(API + path, { headers: { Authorization: 'Bearer ' +
    token } })
    .then(r => r.ok ? r.json() : Promise.reject());

  useEffect(() => {
    if (!isAuthenticated) return;
    api('/orders')
      .then(setOrders)
      .catch(() => logout());
  }, [isAuthenticated]);

  if (!isAuthenticated) return <p>Спершу увійдіть у
систему.</p>;

  if (!orders) return <p>Завантаження...</p>;
  if (orders.length === 0) return <p>Замовлень поки що
немає.</p>;

  return (
    <div style={box}>
      <h2>Історія замовлень</h2>
    </div>
  );
}
```

```

    {orders.map(o => (
      <details key={o.id} style={orderCard}>
        <summary>
          №{o.id} · {new Date(o.created).toLocaleString('uk-
UA')} }

          &nbsp;-&nbsp;
          <strong>{o.total} ₪</strong>
        </summary>

        <ul style={{ marginTop: 8 }}>
          {o.items.map(it => (
            <li key={it.id}>
              {it.title} ×{it.qty} – {it.price * it.qty} ₪
            </li>
          ))}
        </ul>
      </details>
    ))}
  </div>
);
}

/* базові стилі */
const box = { maxWidth: 600, margin: '2rem auto', padding: '0
1rem' };
const orderCard = {
  background: '#fff', padding: '1rem 1.5rem',
  borderRadius: '1rem',
  boxShadow: '0 4px 16px rgba(0,0,0,.1)', marginBottom: '1rem',
  cursor: 'pointer'
};

```

**Програмний код отримане замовлення створене за допомогою форми
на веб-сайті «Маяк»**

```

import React from 'react';
import './OrderReceived.css';

const OrderReceived = ({ order }) => {
  return (
    <section className=«order-received»>
      <h3>Ваше замовлення прийнято!</h3>

      <div className=«order-block»>
        <p><strong>№ замовлення:</strong> {order.id}</p>
        <p><strong>Дата:</strong> {order.date}</p>
        <p><strong>Час:</strong> {order.time}</p>
      </div>

      <div className=«order-block»>
        <p><strong>Ім'я клієнта:</strong> {order.name}</p>
        <p><strong>Телефон:</strong> {order.phone}</p>
        <p><strong>Кількість                гостей:</strong>
{order.persons}</p>
      </div>

      <div className=«order-block»>
        <p><strong>Коментар:</strong></p>
        <p>{order.note || '-'}</p>
      </div>

      <p className=«thanks»>Дякуємо, що обрали «Маяк»! Ми
зв'яжемося з вами для підтвердження.</p>
    </section>
  );
};

export default OrderReceived;

```

**Програмний код онлайн-форми бронювання на веб-сайті ресторану
«Маяк»**

```
import React from 'react';
import './ReservationForm.css';

const ReservationForm = () => {
  return (
    <form className=«reservation-form»>
      <div className=«form-section»>
        <h3>Бронювання столика</h3>
        <label>
          Дата:
          <input type=«date» required />
        </label>
        <label>
          Час:
          <input type=«time» required />
        </label>
        <label>
          Кількість гостей:
          <input type=«number» min=«1» max=«20» required />
        </label>
        <button type=«submit»>Забронювати</button>
      </div>
      <div className=«form-section»>
        <h3>Контактна інформація</h3>
        <label>
          Ім'я:
```



```

        <input type=«text» required />
    </label>
    <label>
        Телефон:
        <input type=«tel» required />
    </label>
    <label>
        Коментар:
        <textarea placeholder=«Особливі побажання»></textarea>
    </label>
</div>
</form>
);
};

```

```
export default ReservationForm;
```