

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

«Вебзастосунок для піцерії»

Виконав студент групи ВТ-21-1
ЗЕМЛЯНСЬКИЙ Нікіта Сергійович

Керівник роботи:
ФАНТ Микола Олександрович

Рецензент:
МОРОЗОВ Андрій Васильович

Житомир – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення
Тетяна Вакалюк
«14» лютого 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ЗЕМЛЯНСЬКИЙ Нікіта Сергійович**

Керівник роботи: **ФАНТ Микола Олександрович**

Тема роботи: **«Вебзастосунок для піцерії»**,

затверджена Наказом закладу вищої освіти від **«14» лютого 2025 р., №61/с**

Вихідні дані для роботи: **Предмет дослідження: методи та інструменти створення сучасних вебзастосунків сфери харчування; Об'єкт дослідження: комплексна реалізація програмного продукту «Вебзастосунок для піцерії»**

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Фант М.О.	27.02.2024р.	06.03.2024р.
2	Фант М.О.	06.03.2024р.	24.04.2024р.
3	Фант М.О.	24.04.2024р.	05.06.2024р.

РЕФЕРАТ

Пояснювальна записка до кваліфікованої роботи на тему «Вебзастосунок для піцерії» складається зі вступу, трьох розділів, висновків, списку використаної літератури та додатків.

Текстова частина викладена на 59 сторінок друкованого тексту. Пояснювальна записка має 8 сторінок додатків. Список використаних джерел містить 30 найменувань і займає 3 сторінки. В роботі наведено 36 рисунків та 10 таблиць. Загальний обсяг роботи - 69 сторінок.

В роботі визначені головні завдання для розробки вебзастосунку для піцерії, проаналізовано аналоги продукту та методи розв'язання поставленої задачі, обґрунтовано вибір архітектури системи.

КЛЮЧОВІ СЛОВА: ВЕБЗАСТОСУНОК, ПІЦЕРІЯ, NEXTJS, PRISMA, POSTGRESQL, БАЗА ДАНИХ, ВЕБРОЗРОБКА.

					ІПЗ.КР.Б-121-25-ПЗ							
Змн.	Арк.	№ докум.	Підпис	Дата								
Розроб.		Землянський Н.С.			Вебзастосунок для піцерії			Лім.	Арк.	Аркушів		
Керівник		Фант М.О.								1		
н. контр.		Тичина О.П.						Житомирська політехніка, група ВТ-21-1				
Зав. каф.		Вакалюк Т.А.										

ABSTRACT

The explanatory note for the qualification work titled "Web Application for a Pizzeria" consists of an introduction, three chapters, conclusions, a list of references, and appendices.

The main textual part is presented in 59 pages of printed text. The explanatory note includes 8 pages of appendices. The list of sources comprises 30 entries and spans 3 pages. The work contains 36 figures and 10 tables. The overall length of the work is 69 pages.

The work identifies the main tasks for developing a web application for a pizzeria, analyzes analogous products and methods for solving the given problem, and substantiates the choice of system architecture.

KEYWORDS: WEB APPLICATION, PIZZERIA, NEXTJS, PRISMA, POSTGRESQL, DATABASE, WEB DEVELOPMENT.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

РЕФЕРАТ	1
ABSTRACT	2
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	4
ВСТУП.....	5
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ВЕБЗАСТОСУНКУ ПІЦЕРІЇ.....	7
1.1. Постановка задачі	7
1.2. Аналіз аналогів програмного продукту	8
1.3. Вибір архітектури вебзастосунок	13
1.4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення	15
Висновки до першого розділу	17
РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКУ ДЛЯ ПІЦЕРІЇ.....	18
2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи.....	18
2.2. Розробка бази даних системи	25
2.3. Проєктування та реалізація алгоритмів роботи вебзастосунок	31
2.4. Реалізація функціоналу вебзастосунок для піцерії	34
Висновки до другого розділу.....	37
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБЗАСТОСУНКОМ ДЛЯ ПІЦЕРІЇ	38
3.1. Порядок встановлення та налаштування параметрів вебзастосунок	38
3.2. Структура інтерфейсу вебзастосунок для піцерії	40
3.3. Тестування роботи вебзастосунок	51
Висновки до третього розділу	53
ВИСНОВКИ	55
ЛІТЕРАТУРА.....	56
ДОДАТКИ.....	59

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

DB – база даних.

CRUD – Create, Read, Update, Delete: операції обробки даних в базі даних.

ORM – технологія, що дозволяє взаємодіяти з базою даних за допомогою об'єктів програмування.

SEO – оптимізація сайту для пошукових систем

SSG – підхід у розробці, при якому сторінки вебсайту генеруються на етапі збірки та зберігаються у вигляді статичних файлів.

SSR – метод рендерингу вебсторінок на сервері перед передачею їх у браузер користувача.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

ВСТУП

Актуальність теми: Сьогодні, у період стрімкого розвитку електронної комерції та зростання попиту на онлайн-сервіси, галузь громадського харчування переживає трансформацію. Однак багато піцерій, особливо локальних, досі використовують застарілі методи роботи: телефонні дзвінки або ручне оформлення замовлень. Це призводить до втрати клієнтів, зниження ефективності та зростання операційних витрат.

Важливим аспектом є також безпека: сучасні технології, такі як NextAuth для аутентифікації та Prisma ORM для роботи з PostgreSQL, забезпечують надійне зберігання персональних даних і платіжної інформації. У контексті зростання кіберзагроз такі рішення стають необхідністю для будь-якого бізнесу.

Враховуючи тенденцію до автоматизації бізнес-процесів, розробка вебзастосунку для піцерії стає критично важливим кроком для підвищення ефективності роботи закладу, покращення якості обслуговування клієнтів та залучення нової аудиторії.

Метою випускної кваліфікованої роботи є закріплення та вдосконалення отриманих навичок та знань з Next.js, TypeScript, Prisma, PostgreSQL, які ляжуть в основу розробки програмного продукту «Вебзастосунок для піцерії».

Результатом виконання випускної роботи стане реалізований вебзастосунок для піцерії, який забезпечить зручний інтерфейс для оформлення замовлень, вибору страв, реєстрації або авторизації. З боку адміністрування, додаток включатиме автоматичні сповіщення для інформування клієнтів, а також функціонал відстеження статусу замовлення.

Для досягнення даної мети та результатів необхідно виконати низку поставлених задач:

- аналіз існуючих рішень для онлайн-замовлень їжі, визначення їхніх переваг і недоліків;
- обґрунтування вибору технологічного стеку;
- проєктування архітектури вебзастосунку;

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

- реалізація основного функціоналу;
- оптимізація продуктивності та адаптація інтерфейсу для різних пристроїв;
- тестування та усунення помилок;

Предметом дослідження є методи та інструменти створення сучасних вебзастосунків сфери харчування.

Об’єктом дослідження є комплексна реалізація програмного продукту «Вебзастосунок для піцерії»

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ВЕБЗАСТОСУНКУ ПІЦЕРІЇ

1.1. Постановка задачі

Метою випускної роботи є розробка вебзастосунку для піцерії, спрямованого на автоматизацію процесу онлайн-замовлень, покращення комунікації з клієнтами та оптимізацію управління бізнес-процесами закладу. Додаток має охоплювати дві основні складові: клієнтську частину та серверну частину.

Процес розробки включає кілька основних етапів. Першим етапом є проектування дизайну інтерфейсу, що починається з визначення загальної концепції. Далі розробляються окремі елементи, такі як кнопки, іконки, поля вводу та меню навігації. Також на цьому етапі визначається кольорова палітра, шрифти, розміри та продумуються механізми взаємодії між елементами. За допомогою графічного редактора Figma створюються макети ключових сторінок, а саме: сторінок входу/реєстрації, головної сторінки, особистого кабінету, профілю користувача, сторінки «Мої замовлення», сторінки товару з детальною інформацією та опціями вибору, а також адміністративної панелі.

Наступним кроком є розробка клієнтської частини (Frontend). Цей етап передбачає імплементацію інтерфейсу з використанням Next.js та TypeScript, реалізацію форм з валідацією, створення адаптивних компонентів за допомогою бібліотеки ShadCN, інтеграцію сповіщень та налаштування роутингу.

Третій етап – це розробка серверної частини (Backend). Він включає налаштування зв'язку з базою даних PostgreSQL через Prisma, реалізацію основних сутностей, створення систем авторизації та аутентифікації, а також налагодження відправки листів на електронну пошту.

Завершальним етапом є тестування, яке охоплює функціональне тестування та тестування графічного інтерфейсу.

Результатом виконання цього плану дій стане створення повноцінного вебзастосунку для піцерії. Готова система буде включати сучасні інструменти

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

безпеки, мати масштабовану архітектуру та зручний інтерфейс, що відповідає актуальним вимогам сучасного ринку.

1.2. Аналіз аналогів програмного продукту

Розглянемо наявні аналоги, які найбільше подібні до теми кваліфікаційної роботи: Okidoki, FichePizza, ChicagoPizza, наведемо їх можливості та порівняємо їх. Основні фактори для порівняння включатимуть зручність використання для користувачів, функціональні можливості, якість обслуговування, підтримку сучасних технологій, швидкість роботи сайту та можливості оплати.

Okidoki. Окрім піци, пропонує широкий асортимент страв японської кухні, зокрема суші, роли, бургери, WOK-страви тощо. Користувачі можуть ознайомитися з меню, замовити доставку кур'єром або обрати самовивіз. Доступ до вебсайту – <https://okidoki.zt.ua/>.

До переваг сервісу можна віднести легкість використання завдяки зрозумілій навігації та мінімалістичному дизайну, а також різноманітний асортимент страв.

Водночас існують і недоліки. Дизайн сайту є застарілим, і на ньому відсутня вбудована платіжна система; хоча безготівкова оплата можлива, перекази потрібно робити вручну на номер картки. Крім того, при розробці були використані неактуальні технології (Yii PHP Framework), відсутня можливість створити особистий кабінет, а швидкість завантаження сторінок та SEO-оптимізація є середніми (рис. 1.2).



					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

Рисунок 1.1 - Головна сторінка Okidoki.

На рисунку 1.1 представлено візуальне оформлення головної сторінки Okidoki. Для оцінки технічних аспектів роботи сайту було проведено аналіз продуктивності, результати якого наведено на рисунку 1.2.

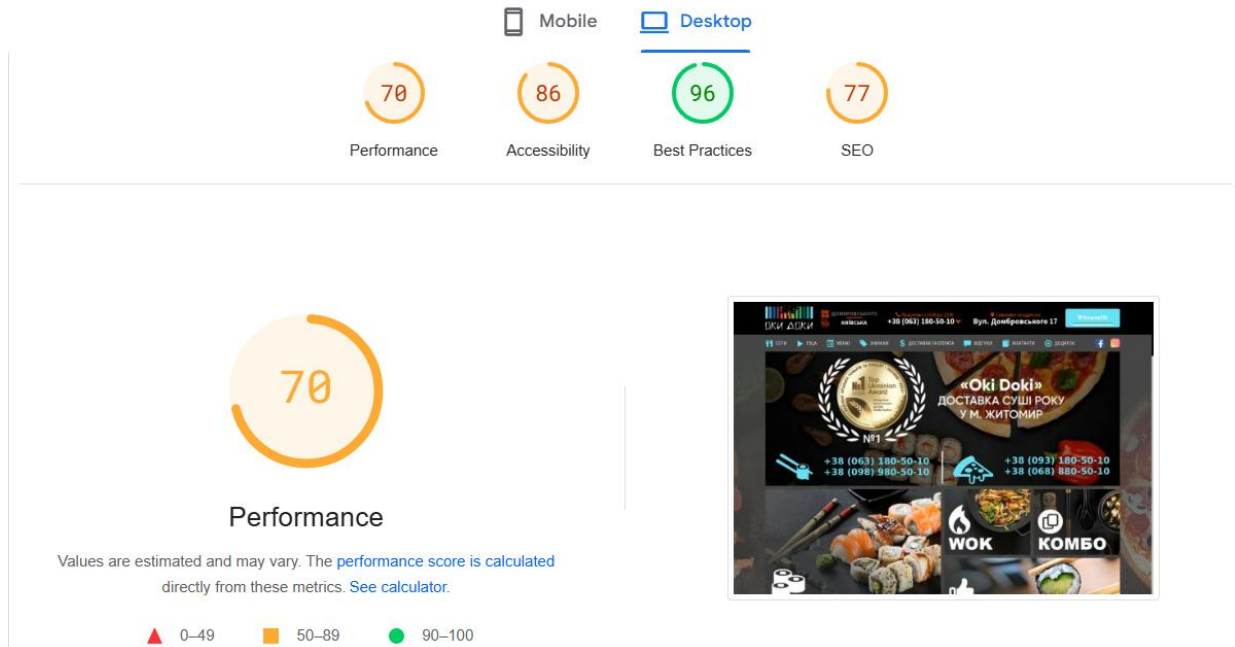


Рисунок 1.2 - Аналіз продуктивності вебсайту Okidoki.

FichePizza. Представляє мережу піцерій Fiche Pizza, яка працює в Україні з 2021 року та має заклади в різних містах, включаючи Коростень, Бердичів, Звягель, Фастів, Софіївську Борщагівку, Вишневе, Святопетрівське, Крюківщину та Житомир. Доступ до вебсайту – <https://fichepizza.com.ua/>.

До переваг сервісу належить сучасний дизайн із чіткою візуальною ієрархією, мінімалістичним стилем та приємною колірною гамою. Користувачам доступний деталізований опис продукту, що включає склад та калорійність, а також зручна безготівкова оплата. Унікальною особливістю є можливість створювати власні піци в конструкторі або додавати інгредієнти до вже існуючих варіантів. Крім того, сайт має високий рейтинг у пошукових системах, розроблений на сучасній платформі OpenCart і демонструє достатньо високий рівень продуктивності (рис. 1.4).

Серед недоліків можна відзначити відсутність особистого кабінету та системи відгуків на сайті.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9



Рисунок 1.3 - Головна сторінка FichePizza.

Інтерфейс сайту FichePizza (Рис. 1.3) є прикладом вдалого поєднання сучасного дизайну та функціональності. Технічний аналіз (Рис. 1.4) показує високі показники продуктивності та SEO.

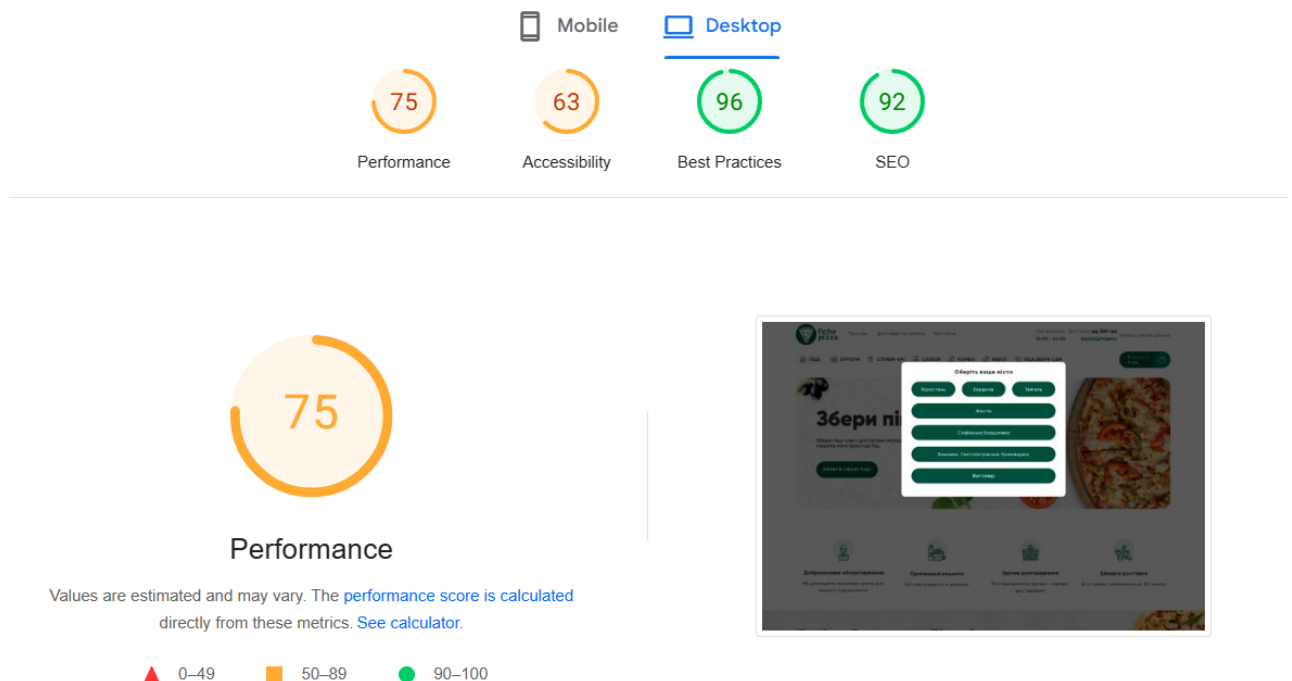


Рисунок 1.4 - Аналіз продуктивності вебсайту FichePizza.

ChicagoPizza. Особливість піцерії досягається завдяки своїй «американській» атмосфері. Відмінність цієї атмосфери полягає у тому, що піци важать близько 1,5 кг, з яких 0,5 кг – це лише сир. Доступ до вебсайту – <https://chicagopizza.com/>.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

До переваг можна віднести креативний та цікавий дизайн сайту, де використано контрастні кольори для передачі атмосфери, а також зручну візуалізацію зон доставки через інтегровану мапу та можливість безготівкової оплати.

Однак, існують і критичні недоліки. На сайті присутні мовні помилки у вигляді некоректного перекладу деяких слів українською. Також спостерігаються технічні збої в роботі кошика, зокрема повільний редірект на головну сторінку при додаванні декількох піц. Критично низька продуктивність вебсервісу зумовлена неоптимізованими зображеннями, анімаціями та застарілим кодом. Крім того, на сайті відсутня система відгуків та особистий кабінет для користувачів, а технологічний стек є застарілим і не оновлюється (WordPress 6.1.1, WooCommerce 4.3.0).



Рисунок 1.5 - Головна сторінка ChicagoPizza.

Головна сторінка ChicagoPizza (Рис. 1.5) має креативний дизайн, проте її технічна реалізація, як свідчить аналіз продуктивності (Рис. 1.6), має суттєві

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

недоліки.

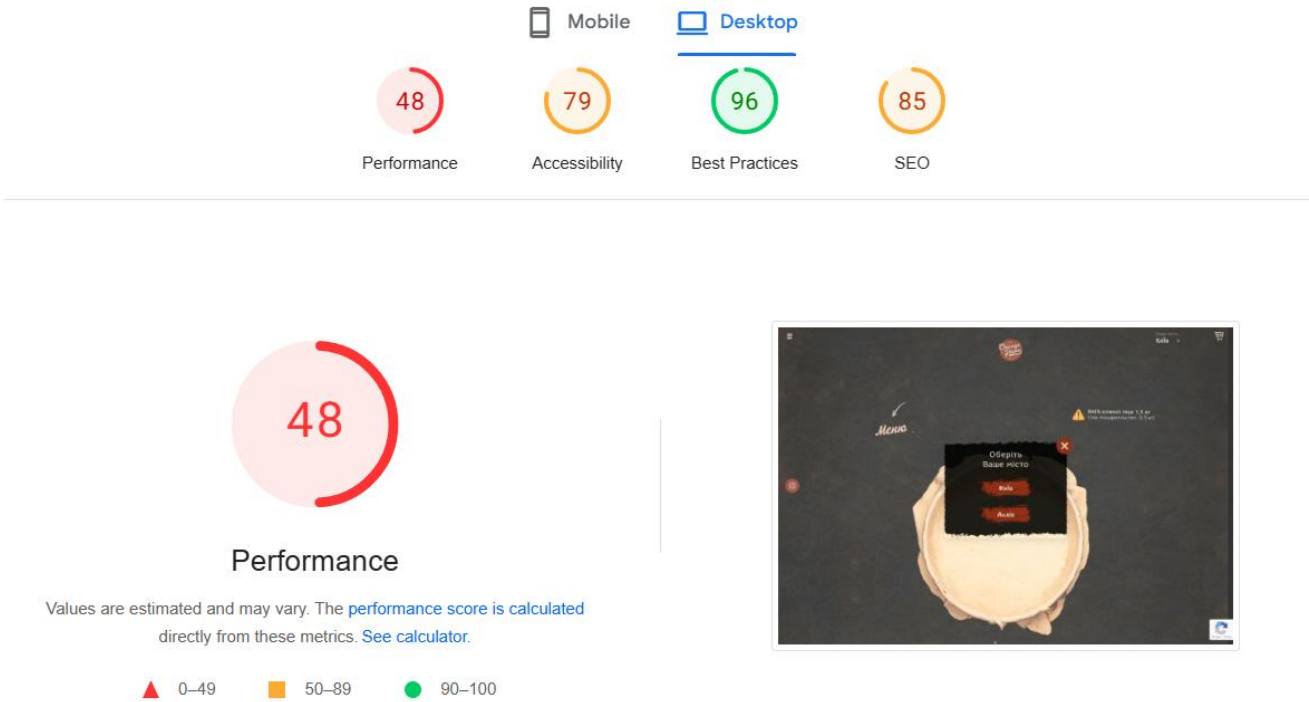


Рисунок. 1.6 - Аналіз продуктивності вебсайту ChicagoPizza.

У таблиці 1.1 наведено порівняльну характеристику проаналізованих вебсайтів-аналогів за ключовими функціональними та технічними параметрами:

Таблиця 1.1

Порівняння вебсайтів

Вебсайти	Okidoki	Fichepizza	ChicagoPizza
Сучасний дизайн	-	+	+
Наявність основних функцій	+	+	+
Вбудована безготівкова оплата	-	+	+
Особистий кабінет	-	-	-
Додаткові можливості (конструктор піц)	-	+	-
Використання сучасних технологій	-	+	-
Продуктивність сайту	+	+	-
SEO-оптимізація	-	+	-

Аналіз вебплатформ Okidoki, FichePizza та ChicagoPizza дозволив виокремити ключові вимоги до майбутнього вебзастосунку для піцерії.

Першочерговою вимогою є створення інтуїтивного інтерфейсу з чіткою навігацією та адаптивним дизайном.

Для ефективного управління бізнес-процесами передбачено адміністративну панель, яка дасть змогу контролювати меню, обробляти замовлення та аналізувати продажі.

Вебзастосунок передбачає три основні ролі. Гості отримують доступ до перегляду меню та можливості реєстрації. Зареєстровані користувачі можуть оформляти замовлення, керувати особистим кабінетом і залишати відгуки. Адміністратори матимуть повний контроль над контентом, зможуть аналізувати статистику та керувати замовленнями через спеціальну панель.

1.3. Вибір архітектури вебзастосунку

Для реалізації вебзастосунку піцерії було обрано Jamstack-подібну [5] архітектуру. Next.js поєднує клієнтську та серверну логіку в єдиному стеку [13]. Цей підхід дозволяє ефективно інтегрувати REST API безпосередньо в Next.js за допомогою вбудованих API-маршрутів [12], що спрощує розробку та підтримку системи.

Вибір саме цієї архітектури був зумовлений кількома важливими факторами. По-перше, монолітна архітектура Next.js дозволяє розробникам працювати з єдиним кодовим репозиторієм, що значно спрощує процес розробки та тестування. Замість того, щоб підтримувати окремі сервери для клієнтської та серверної частини, вся логіка зосереджується в одному проекті. Це особливо важливо для середніх за розміром проектів, таких як вебзастосунок піцерії, де складність не виправдовує використання мікросервісної архітектури.

Другим вагомим аргументом на користь обраної архітектури є можливість серверного рендерингу. Next.js надає вбудовану підтримку Server-Side Rendering (SSR) та Static Site Generation (SSG), що критично важливо для сайтів комерційного призначення. Серверний рендеринг забезпечує швидке завантаження сторінок, що безпосередньо впливає на користувацький досвід та конверсію замовлень. Клієнти піцерії очікують швидкого доступу до меню та можливості оперативного

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

оформлення замовлень, тому продуктивність інтерфейсу має першочергове значення.

Архітектура складається з кількох ключових компонентів. Клієнтська частина (Frontend) реалізована за допомогою React і TypeScript, відповідає за відображення інтерфейсу користувача. У свою чергу серверна частина (Backend) обробляє бізнес-логіку, взаємодію з базою даних через Prisma ORM [18] та автентифікацію. Для надійного зберігання даних про користувачів, замовлення та страви використовується база даних PostgreSQL.

На рисунку 1.7 наведено загальний принцип роботи обраної архітектури, де показано взаємодію між основними компонентами системи та потік даних від користувача до бази даних.

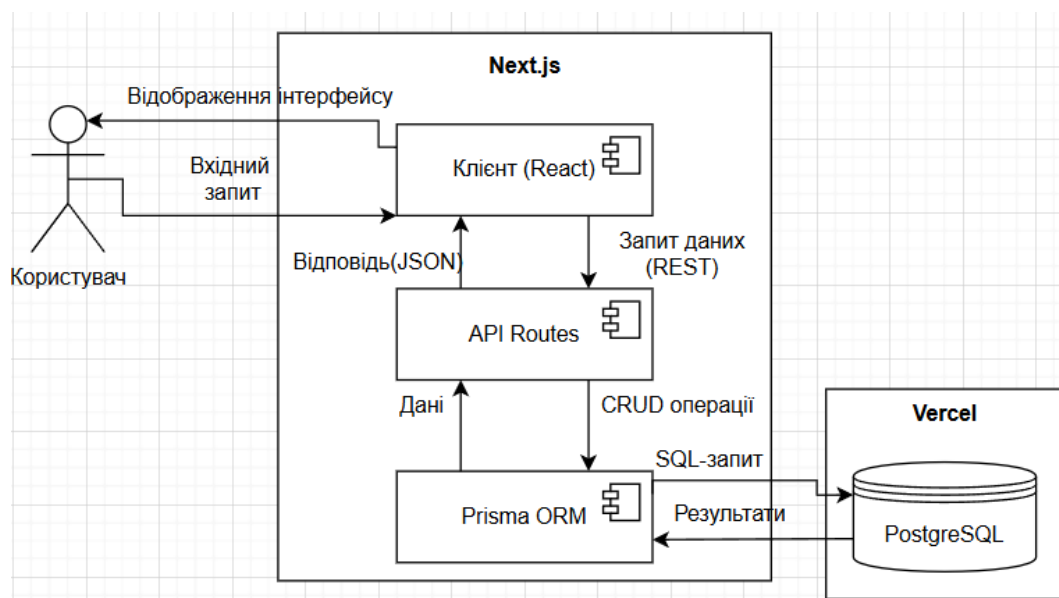


Рисунок 1.7 - Загальний принцип роботи архітектури.

Цикл обробки запиту реалізовано наступним чином. Спочатку користувач взаємодіє з клієнтською частиною, наприклад, додаючи піцу до кошика. Після цього Next.js обробляє цей запит через відповідний API-маршрут, який, у свою чергу, викликає необхідні функції Prisma для роботи з базою даних. Далі Prisma ORM виконує CRUD-операції, отримуючи або змінюючи інформацію безпосередньо в базі даних. На наступному етапі серверна частина перевіряє права доступу користувача та відправляє відповідь назад клієнту. На завершення циклу

клієнтська частина оновлює інтерфейс на основі отриманих даних, наприклад, відображаючи підтвердження замовлення.

Такий підхід сприяє оптимізації розробки, забезпечує високу продуктивність і масштабованість, а також гарантує безпеку системи. Використання зазначеного стеку технологій дозволяє ефективно задовольнити вимоги як кінцевих користувачів, так і адміністраторів вебзастосунку для піцерії.

1.4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Вибір технологічного стеку для розробки вебзастосунку для піцерії був ретельно продуманим процесом, що ґрунтувався на потребі створити сучасне, масштабоване, безпечне та продуктивне рішення. Кожен інструмент обирався не ізольовано, а як частина єдиної екосистеми, де технології доповнюють одна одну, забезпечуючи ефективність як на етапі розробки, так і під час експлуатації продукту.

Основою всього проєкту став фреймворк Next.js, який дозволив реалізувати як клієнтську так і серверну частини в межах одного проєкту. Такий підхід значно зменшує складність розробки, оскільки немає потреби налаштовувати взаємодію між двома окремими застосунками. Next.js, побудований на базі бібліотеки React, надає потужні можливості для створення швидких та інтерактивних користувацьких інтерфейсів. Водночас завдяки вбудованим механізмам, таким як рендеринг на стороні сервера (SSR) та генерація статичних сайтів (SSG), він забезпечує високу швидкість завантаження сторінок та чудову SEO-оптимізацію, що є критично важливим для залучення клієнтів через пошукові системи.

Для підвищення надійності та якості коду було використано TypeScript. Ця мова додає до JavaScript строгу типізацію, що дозволяє виявляти та виправляти багато потенційних помилок ще на етапі написання коду, а не після того, як їх знайдуть користувачі. Це не тільки покращує стабільність застосунку, але й робить кодову базу більш зрозумілою та легкою для підтримки в майбутньому.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

Ключовим елементом для роботи з даними стала Prisma ORM, яка виступає як посередник між програмним кодом та базою даних. Технологія ORM дозволяє взаємодіяти з базою даних за допомогою об'єктів TypeScript, замість написання складних SQL-запитів вручну. Prisma надає сучасний та безпечний підхід до роботи з даними, забезпечуючи автоматичну генерацію типів та підтримку міграцій, що значно спрощує процес оновлення структури бази даних. Як система управління базами даних було обрано PostgreSQL [16] - надійне та перевірене часом рішення, відоме своєю стабільністю, підтримкою складних транзакцій та здатністю ефективно обробляти великі обсяги інформації, що є необхідним для зберігання даних про клієнтів, замовлення та меню.

Для розгортання та хостингу вебзастосунку було обрано платформу Vercel. Цей вибір обумовлений її тісною інтеграцією з Next.js, оскільки Vercel розроблена тією ж компанією [25]. Платформа забезпечує автоматичне розгортання, глобальну мережу доставки контенту (CDN) та безсерверне середовище, що гарантує високу доступність та швидкість роботи застосунку для користувачів у будь-якій точці світу. Крім того, наявність безкоштовного тарифного плану для некомерційних та стартових проектів робить її ідеальним вибором для цього проекту.

Візуальна складова та користувацький досвід є не менш важливими, тому для створення інтерфейсу було використано CSS-фреймворк TailwindCSS та бібліотеку компонентів ShadCN [21]. TailwindCSS дозволяє швидко створювати сучасні та адаптивні дизайни завдяки своєму підходу "utility-first", що значно прискорює процес версії. ShadCN, у своєму роді, надає набір готових, стилізованих та доступних компонентів інтерфейсу, що дозволяє зосередитись на функціональності, а не на розробці базових елементів з нуля.

Вимоги до апаратного забезпечення:

- Клієнтська частина працює в сучасних браузерах (Chrome, Firefox, Safari) і не вимагає потужних пристроїв. Достатньо 2 ГБ оперативної пам'яті та підтримки ES6+ для JavaScript.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

- Серверна частина розгортається на Vercel, який надає необхідні ресурси для роботи Next.js та PostgreSQL. Мінімальні вимоги включають 512 MB оперативної пам'яті та 1 vCPU.

Таким чином, використання вказаних інструментальних засобів створює єдину, злагоджену систему. Вона забезпечує надійну та ефективну обробку даних, а хостинг на Vercel дозволяє швидко та з мінімальними витратами розгортати додаток. Поєднання ж TailwindCSS і ShadCN дає змогу реалізувати привабливий та адаптивний інтерфейс, що відповідає сучасним вимогам користувачів.

Висновки до першого розділу

У даному розділі було проведено комплексний аналіз сучасних вебзастосунків для піцерій, який виявив обмежену функціональність та застарілі технології в більшості аналогів, що обґрунтувало необхідність розробки сучасного рішення.

Основою архітектури обрано Jamstack-підхід із Next.js, який поєднує клієнтську та серверну логіку в єдиному стеку з інтегрованим REST API. Використання Prisma ORM та PostgreSQL забезпечило надійну взаємодію з базою даних, автоматизацію міграцій і типізацію запитів для безпеки та масштабованості.

Вибір інструментів обґрунтовано потребою створення сучасного рішення: Next.js забезпечує гібридний рендеринг для покращення SEO, TypeScript додає строгу типізацію, TailwindCSS і ShadCN прискорюють розробку інтерфейсу, а хостинг на Vercel надає інтеграцію з Next.js та автоматичне масштабування.

Апаратні вимоги оптимізовано для стартових проєктів через використання мінімальних ресурсів Vercel, при цьому клієнтська частина працює на сучасних браузерах без високих вимог до продуктивності пристроїв. Цей розділ став основою для подальшої реалізації вебзастосунку, визначивши архітектуру системи, інструменти та технічні вимоги.

В результаті, цей розділ став основою для подальшої реалізації вебзастосунку, провівши аналіз аналогічних платформ, вибравши архітектуру системи, інструментів і апаратних вимог.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКУ ДЛЯ ПІЦЕРІЇ

2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи

Вебзастосунок для піцерії надає інтуїтивно зрозумілий онлайн-інтерфейс, через який користувачі можуть переглядати меню, налаштовувати піци, оформлювати замовлення та керувати ними, тоді як адміністратори мають доступ до інструментів для ефективного управління бізнес-процесами закладу. Функціональність системи залежить від активності різних груп користувачів.

Ролі можуть бути розділені на три основні групи: Гості, які переглядають меню та роблять замовлення без обов'язкової реєстрації; Залогінені користувачі, які мають доступ до персоналізованих функцій, таких як історія замовлень та керування профілем; та Адміністратори, що відповідають за управління контентом, замовленнями та аналітикою.

Для наглядної взаємодії між елементами застосунку було створено діаграму варіантів використання (рис 2.1). Дана діаграма, як і подальші моделі в цьому підрозділі, розроблені з використанням нотації UML (Unified Modeling Language) [1]. Як інструмент для їх створення було обрано draw.io [2].

Складники діаграми варіантів використання будуть виглядати наступним чином:

У системі передбачена можливість «Переглянути меню», якою можуть скористатися як Гості, так і Залогінені користувачі. Це дозволяє їм ознайомитися з асортиментом доступних продуктів, прочитати їх опис та дізнатися актуальні ціни.

Для Гостей та Залогінених користувачів також доступна функція «Налаштувати піцу». Вона дає змогу не просто обрати стандартний варіант, а й підібрати різні варіації піци, а також додати до неї ті інгредієнти, які найбільше до вподоби.

Після того, як вибір зроблено, як Гості, так і Залогінені користувачі можуть «Додати товар до кошика». Ця дія дозволяє зібрати всі обрані страви, включаючи

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		18

індивідуально налаштовані піци, у віртуальний кошик для подальшого переходу до оформлення замовлення.

Наступним кроком для Гостей та Залогієних користувачів є «Оформити замовлення». На цьому етапі користувачі фіналізують свою покупку, надаючи всю необхідну контактну інформацію та деталі для доставки.

Гості системи мають можливість «Зареєструватися». Процес реєстрації передбачає створення нового облікового запису, що в подальшому відкриває доступ до розширених та персоналізованих функцій вебзастосунку.

Щоб отримати доступ до свого облікового запису, Залогієні користувачі та Адміністратори використовують функцію «Увійти в систему». Це стандартна процедура, яка вимагає введення персональних облікових даних для автентифікації в системі.

Для Залогієних користувачів передбачена можливість «Керувати особистим кабінетом». У своєму кабінеті вони можуть переглядати збережені персональні дані, а також за потреби вносити зміни та оновлювати їх.

Також Залогієні користувачі можуть «Переглянути історію замовлень». Ця функція дозволяє їм не тільки бачити список усіх попередньо зроблених замовлень, але й відслідковувати поточний статус виконання кожного з них.

Адміністратор системи має ексклюзивні права на «Керувати товарами». Це означає, що він може додавати до меню нові страви та інгредієнти, редагувати інформацію про вже існуючі позиції, а також, за необхідності, видаляти їх з асортименту.

Окрім товарів, Адміністратор також може «Керувати категоріями товарів». Ця функція дозволяє йому створювати нові категорії для страв у меню, що полегшує навігацію для користувачів, а також редагувати назви чи видаляти існуючі категорії.

Важливою функцією Адміністратора є можливість «Змінити статус замовлення». Він може оновлювати поточний стан замовлення, наприклад, позначаючи його як «Готується», «В дорозі» чи «Доставлено», що інформує клієнта про процес виконання.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

Нарешті, Адміністратор має доступ до функції «Переглянути аналітику». Цей інструмент надає йому доступ до різноманітних статистичних даних, які можуть стосуватися обсягів продажів, популярності окремих страв, загальної кількості замовлень та інших важливих показників роботи піцерії.



Рисунок 2.1 - Діаграма варіантів використання.

Основна ціль розробки вебзастосунку для піцерії полягає в створенні онлайн-середовища, яке дозволить автоматизувати процес приймання та обробки замовлень, покращити комунікацію з клієнтами та оптимізувати управління бізнес-процесами закладу. Платформа спрямована на забезпечення зручного та

ефективного механізму для вибору страв, їх налаштування та оформлення доставки або самовивозу. Основна мета полягає у сприянні зростанню ефективності роботи піцерії, підвищенню якості обслуговування клієнтів та залученню нової аудиторії. Застосунок надає можливість клієнтам легко ознайомлюватися з меню, робити усвідомлений вибір та відстежувати статус своїх замовлень, а адміністраторам – ефективно керувати асортиментом та операціями.

Вимоги користувачів

Для внутрішніх користувачів, зокрема адміністратора, система повинна забезпечувати можливість керування меню, що включає додавання, редагування та видалення страв і інгредієнтів. Адміністратор повинен мати доступ до перегляду та зміни статусу замовлень, а також можливість переглядати аналітичні дані, які охоплюють інформацію про продажі, популярність окремих страв і загальну статистику замовлень. Також важливим є забезпечення функцій керування користувачами, зокрема видалення облікових записів.

Залогінений клієнт повинен мати змогу проходити авторизацію, переглядати меню та ціни, налаштовувати піцу за власним бажанням шляхом вибору інгредієнтів, розміру й типу тіста. Окрім цього, користувач може додавати товари до кошика, оформлювати замовлення, переглядати історію замовлень та їх статус, керувати особистими даними у своєму профілі, а також отримувати сповіщення про зміну статусу замовлення.

Гість, тобто неавторизований користувач, також має змогу переглядати меню та ціни, налаштовувати піцу, додавати товари до кошика й оформлювати замовлення без обов'язкової реєстрації. За потреби, йому надається можливість створити новий обліковий запис.

Функціональні вимоги

Система повинна забезпечувати можливість перегляду меню з детальною інформацією про кожну страву, включно з описом і ціною. Вона повинна дозволяти налаштовувати піцу шляхом вибору розміру, типу тіста й додаткових інгредієнтів. Також необхідно реалізувати функціонал кошика для керування доданими

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

товарами, зокрема можливість змінювати їх кількість або видаляти. Система повинна підтримувати процес оформлення замовлення.

Для ідентифікації користувачів повинна бути реалізована реєстрація та авторизація через електронну пошту й пароль. Адміністративна панель має надавати інструменти для керування стравами, інгредієнтами, категоріями меню, замовленнями та користувачами. Клієнтам повинна відображатися інформація про поточний статус їх замовлень. Окрім цього, адміністраторам необхідно мати доступ до аналітики та звітів про продажі та ефективність роботи сервісу. Система повинна підтримувати інтеграцію з платіжними системами, а також мати можливість надсилати користувачам сповіщення про зміну статусу замовлення.

Нефункціональні вимоги

Щодо сприйняття, інтерфейс вебсайту повинен бути інтуїтивно зрозумілим та зручним, не вимагати додаткової підготовки, а освоєння його основних функцій не повинно перевищувати трьох-п'яти хвилин (для досвідчених користувачів – менше двох хвилин). Вебсайт має реагувати на дії користувача не довше трьох секунд.

З точки зору надійності, вебсайт повинен бути доступний упродовж усієї доби. Продуктивність системи повинна дозволяти одночасну роботу великої кількості користувачів без суттєвого зниження швидкодії.

Безпека є критично важливою: система повинна захищати персональні дані користувачів згідно з чинним законодавством [29], а також забезпечувати безпеку платіжних даних і транзакцій [15].

Системні вимоги

Виконання вебсайту має бути можливим на пристроях, що відповідають мінімальній конфігурації: 2 ГБ оперативної пам'яті, процесор із тактовою частотою не нижче 1.4 ГГц, доступ до інтернету та наявність сучасного веббраузера, такого як Google Chrome, Mozilla Firefox, Microsoft Edge або Safari останніх версій.

Інформаційна система повинна використовувати реляційну базу даних PostgreSQL для зберігання даних про користувачів, замовлення та меню. Серверна

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

частина повинна підтримувати технології Node.js для бекенду та Next.js для рендерингу інтерфейсу користувача.

Після аналізу функціональних вимог вдалося визначити ключові сутності, які необхідні для програмної реалізації застосунку. На рисунку 2.2 подано UML-діаграму класів, яка використовується для його роботи.

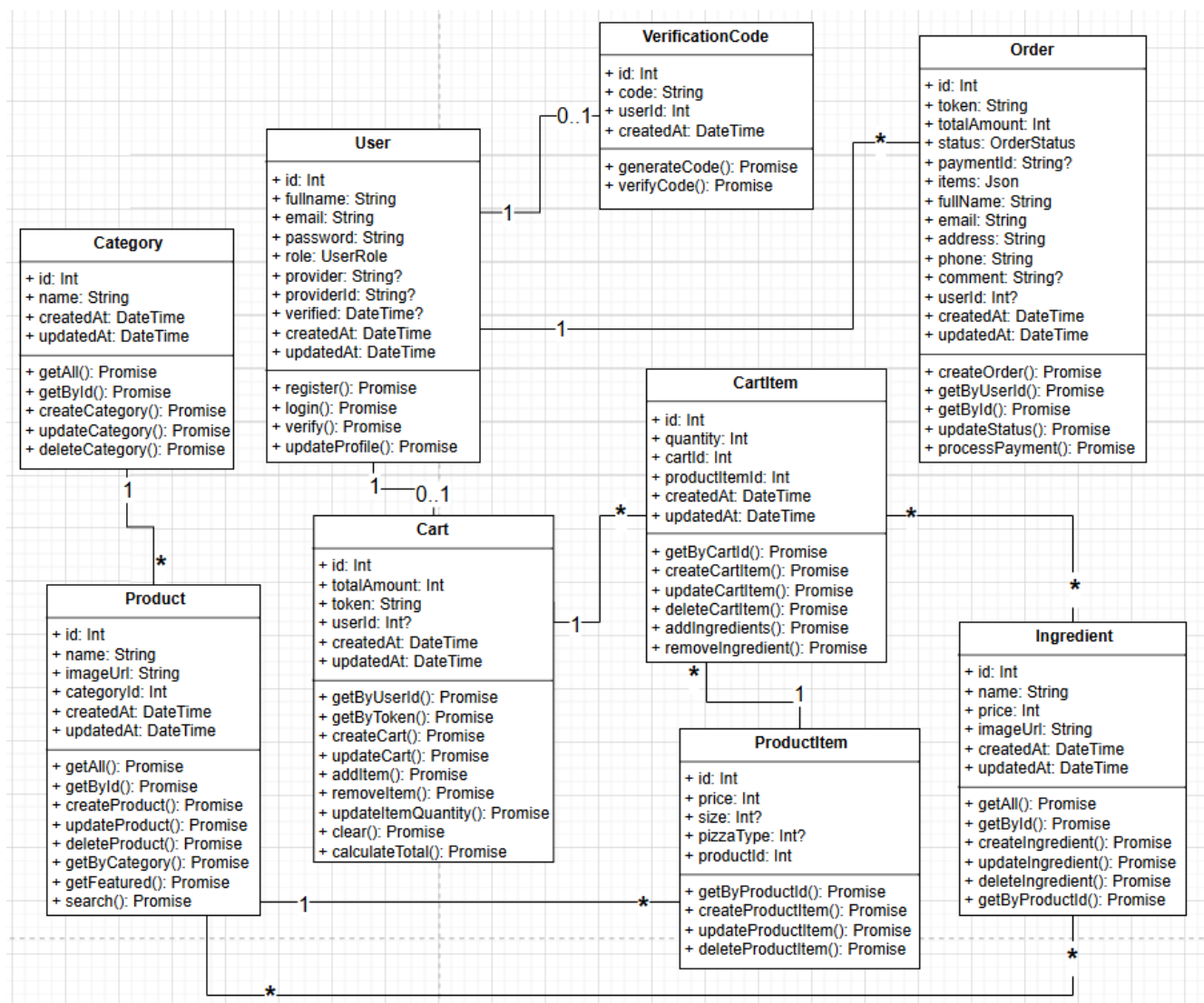


Рисунок 2.2 - Діаграма класів.

Нижче наведено детальний опис діаграми класів, що розкриває функціональність методів для кожної сутності.

Сутність User представляє користувача системи та включає методи для реєстрації нового користувача (register()), авторизації (login()), верифікації облікового запису (verify()) та редагування профілю (updateProfile()).

Сутність Category відповідає за категорії продуктів і дозволяє отримувати всі категорії (getAll()) або конкретну категорію за її ідентифікатором (getById()). Також передбачені функції для створення (createCategory()), оновлення (updateCategory()) та видалення (deleteCategory()) категорій.

Для представлення страви або товару в меню використовується сутність Product. Її методи дозволяють отримати список усіх продуктів (getAll()), конкретний продукт за ідентифікатором (getById()), список продуктів за певною категорією (getByCategory()), а також здійснювати пошук за ключовим словом (search()). Крім того, реалізовано можливість створювати (createProduct()), оновлювати (updateProduct()) та видаляти (deleteProduct()) продукти.

Сутність Ingredient описує додаткові інгредієнти, які можна додати до продукту. Вона містить методи для отримання повного списку інгредієнтів (getAll()), конкретного інгредієнта за ідентифікатором (getById()), а також списку доступних інгредієнтів для певного продукту (getByProductId()). Передбачено також створення (createIngredient()), оновлення (updateIngredient()) та видалення (deleteIngredient()) інгредієнтів.

Конкретний варіант продукту, наприклад, піца різного розміру або типу тіста, представляє сутність ProductItem. Вона дозволяє отримувати всі варіації продукту за його ідентифікатором (getByProductId()), а також створювати (createProductItem()), оновлювати (updateProductItem()) та видаляти (deleteProductItem()) такі варіанти.

Сутність Cart представляє кошик покупок користувача. Вона надає можливість отримати кошик за ідентифікатором користувача (getByUserId()) або за токеном (getByToken()). Методи цієї сутності дозволяють створювати (createCart()) та оновлювати (updateCart()) кошик, додавати (addItem()) та видаляти (removeItem()) товари, оновлювати кількість товару (updateItemQuantity()), повністю очищувати кошик (clear()) та розраховувати загальну суму (calculateTotal()).

Окремий товар у кошику описується сутністю CartItem. Вона дозволяє отримувати товари за ідентифікатором кошика (getByCartId()), створювати

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

(createCartItem()), оновлювати (updateCartItem()) та видаляти (deleteCartItem()) товари в кошику. Також є методи для додавання (addIngredients()) та видалення (remove Ingredient()) додаткових інгредієнтів до конкретного товару в кошику.

Зроблене користувачем замовлення представляє сутність Order. Її методи включають створення замовлення (createOrder()), отримання замовлення за ідентифікатором (getById()), оновлення його статусу (updateStatus()) та обробку платежу (process Payment()).

Нарешті, сутність VerificationCode відповідає за код для верифікації електронної пошти, маючи методи для генерації (generateCode()) та перевірки (verifyCode()) цього коду.

2.2. Розробка бази даних системи

Для забезпечення ефективного зберігання, управління та доступу до даних вебзастосунку для піцерії було спроектовано реляційну базу даних. Вибір системи управління базами даних (СУБД) PostgreSQL обумовлений її надійністю, гнучкістю, підтримкою складних запитів та можливістю масштабування, що є важливими факторами для вебзастосунку. Для взаємодії з базою даних та визначення її схеми використовується Prisma ORM[17], що дозволяє описувати моделі даних мовою TypeScript [24] та автоматично генерувати міграції бази даних.

Структура бази даних розроблена на основі об'єктно-орієнтованої моделі, описаної в попередньому розділі (рис. 2.2), і включає набір взаємопов'язаних таблиць, які відображають ключові сутності системи (рис. 2.3).

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

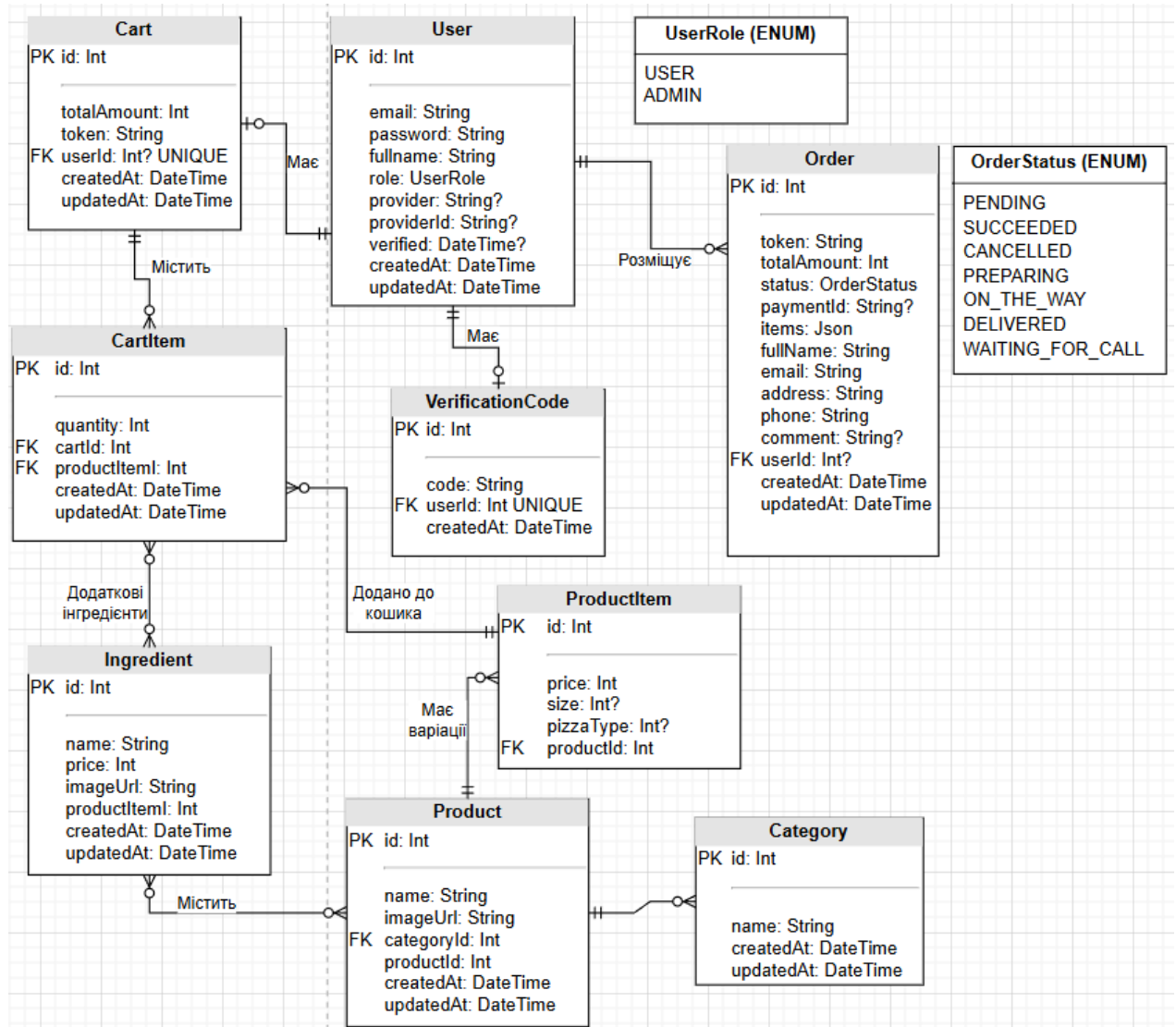


Рисунок 2.3 - Діаграма бази даних.

База даних системи складається з кількох основних таблиць, кожна з яких виконує свою функцію. Таблиця User зберігає інформацію про зареєстрованих користувачів, тоді як Category містить дані про категорії товарів, такі як "Піца", "Напої" та "Десерти". Конкретні товари, що пропонуються в піцерії, описані в таблиці Product, а таблиця Ingredient містить перелік доступних інгредієнтів та їхні характеристики.

Для деталізації різних варіантів одного продукту, наприклад, розміру піци або типу тіста, використовується таблиця ProductItem. Інформація про кошики покупок користувачів зберігається в таблиці Cart, а CartItem деталізує товари, які були додані до цих кошиків. Дані про всі оформлені замовлення знаходяться в

таблиці Order. Нарешті, таблиця VerificationCode використовується для зберігання кодів, необхідних для підтвердження електронної пошти.

Також використовуються enum-перерахування, зокрема UserRole та OrderStatus, що дозволяє фіксувати ролі користувачів (USER, ADMIN) та статуси замовлення (PENDING, DELIVERED, тощо).

Опис полів кожної таблиці бази даних наведено в таблицях 2.1 – 2.9.

Таблиця 2.1

Опис таблиці User

Поле	Тип	Призначення
id	Int (PRIMARY KEY)	Унікальний ідентифікатор користувача
fullName	String	Повне ім'я користувача
email	String(UNIQUE)	Електронна пошта користувача
password	String	Хешований пароль користувача
role	UserRole(ENUM)	Роль користувача в системі (USER, ADMIN), за замовчуванням USER
provider	String?	Назва провайдера соціальної аутентифікації
providerId	String?	Ідентифікатор користувача у провайдера соціальної аутентифікації, необов'язкове
verified	DateTime?	Дата підтвердження облікового запису
createdAt	DateTime	Дата та час створення запису
updatedAt	DateTime	Дата та час останнього оновлення запису

Таблиця 2.2

Опис таблиці Category

Поле	Тип	Призначення
id	Int (PRIMARY KEY)	Унікальний ідентифікатор категорії

Продовження таблиці 2.2

name	String	Назва категорії
createdAt	DateTime	Дата та час створення запису
updatedAt	DateTime	Дата та час останнього оновлення запису

Таблиця 2.3

Опис таблиці Product

Поле	Тип	Призначення
id	Int (PRIMARY KEY)	Унікальний ідентифікатор продукту
name	String	Назва продукту
imageUrl	String	URL-адреса зображення продукту
categoryId	Int	Ідентифікатор категорії, до якої належить продукт
createdAt	DateTime	Дата та час створення запису
updatedAt	DateTime	Дата та час останнього оновлення запису

Таблиця 2.4

Опис таблиці Ingredient

Поле	Тип	Призначення
id	Int (PRIMARY KEY)	Унікальний ідентифікатор інгредієнта
name	String	Назва інгредієнта
price	Int	Ціна інгредієнта
imageUrl	String	URL-адреса зображення інгредієнта
createdAt	DateTime	Дата та час створення запису
updatedAt	DateTime	Дата та час останнього оновлення запису

Таблиця 2.5

Опис таблиці ProductItem

Поле	Тип	Призначення
id	Int (PRIMARY KEY)	Унікальний ідентифікатор варіанта продукту
price	Int	Ціна варіанта продукту
size	Int?	Розмір
pizzaType	Int?	Тип піци
productId	Int (FOREIGN KEY)	Ідентифікатор основного продукту, до якого належить

Таблиця 2.6

Опис таблиці Cart

Поле	Тип	Призначення
id	Int (PRIMARY KEY)	Унікальний ідентифікатор кошика
totalAmount	Int	Загальна сума товарів у кошику, за замовчуванням
token	String	Унікальний токен для ідентифікації кошика
userId	Int? (FOREIGN KEY)	Ідентифікатор користувача, якому належить кошик
createdAt	DateTime	Дата та час створення запису
updatedAt	DateTime	Дата та час останнього оновлення запису

Таблиця 2.7

Опис таблиці CartItem

Поле	Тип	Призначення
------	-----	-------------

id	Int (PRIMARY KEY)	Унікальний ідентифікатор елемента кошика
quantity	Int	Кількість одиниць товару в кошику
cartId	Int (FOREIGN KEY)	Ідентифікатор кошика, до якого належить елемент
productId	Int (FOREIGN KEY)	Ідентифікатор варіанта продукту, що доданий до кошика
createdAt	DateTime	Дата та час створення запису
updatedAt	DateTime	Дата та час останнього оновлення запису

Таблиця 2.8

Опис таблиці Order

Поле	Тип	Призначення
id	Int (PRIMARY KEY)	Унікальний ідентифікатор замовлення
token	String	Унікальний токен для ідентифікації замовлення
totalAmount	Int	Загальна сума замовлення
status	OrderStatus (ENUM)	Поточний статус замовлення
paymentId	String?	Ідентифікатор транзакції платежу
items	Json	Деталізована інформація про замовлені товари у форматі JSON
fullName	String	Повне ім'я отримувача
email	String	Електронна пошта отримувача
address	String	Адреса доставки
phone	String	Контактний телефон отримувача
comment	String?	Коментар до замовлення

userId	Int? (FOREIGNKEY)	Ідентифікатор користувача, який зробив замовлення
createdAt	DateTime	Дата та час створення запису
updatedAt	DateTime	Дата та час останнього оновлення запису

Таблиця 2.9

Опис таблиці VerificationCode

Поле	Тип	Призначення
id	Int (PRIMARY KEY)	Унікальний ідентифікатор коду
code	String	Сам верифікаційний код
userId	Int (FOREIGN KEY)	Ідентифікатор користувача, для якого призначений код
createdAt	DateTime	Дата та час створення запису

2.3. Проектування та реалізація алгоритмів роботи вебзастосунку

Ефективна робота вебзастосунку для піцерії залежить від чітко визначених та коректно реалізованих алгоритмів, що описують взаємодію користувачів із системою та обробку даних на серверній стороні. На основі визначених функціональних вимог та варіантів використання були спроектовані ключові алгоритми, що охоплюють основні сценарії роботи користувачів: від вибору страв до оформлення та відстеження замовлення, а також адміністративні функції.

Для візуалізації загальної логіки взаємодії користувача з вебзастосунком розроблено діаграму активності (рис. 2.4). Вона відображає основні стани та переходи, які може здійснювати користувач, залежно від його ролі та дій.

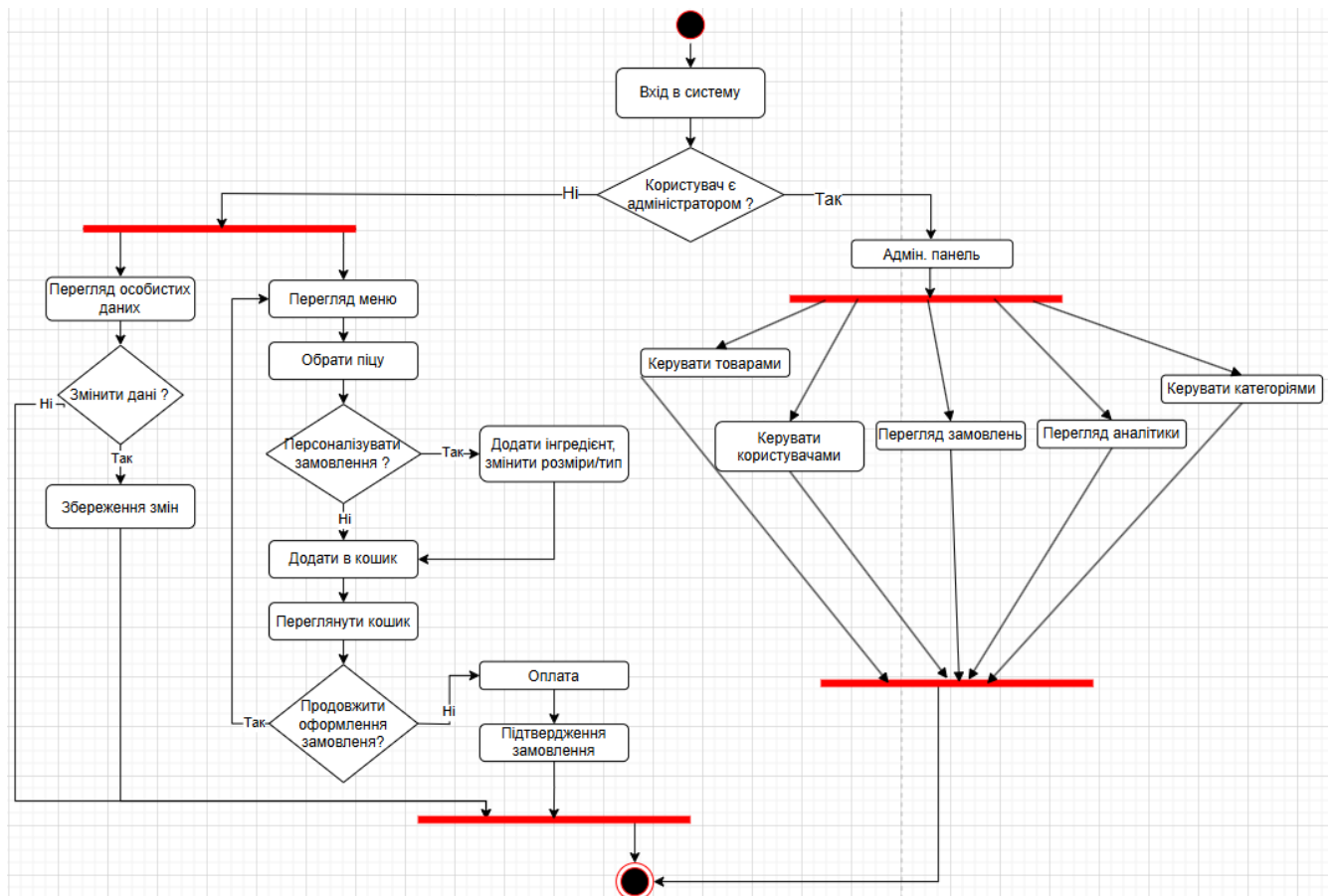


Рисунок 2.4 - Діаграма активності користувача в системі.

Залогінений користувач: Після успішної автентифікації користувач отримує доступ до особистого кабінету, де може керувати своїми даними, переглядати історію та статуси своїх замовлень. Процес вибору товарів та оформлення замовлення аналогічний до гостьового, але спрощений за рахунок використання збереженої інформації.

Адміністратор: Після входу до адміністративної панелі адміністратор отримує доступ до інструментів управління контентом (товари, категорії, інгредієнти), обробки замовлень, перегляду статистики та управління обліковими записами користувачів.

Алгоритм оформлення замовлення

Одним із центральних алгоритмів системи є процес оформлення замовлення. Він включає низку кроків від формування кошика до підтвердження замовлення. Діаграма послідовності (рис. 2.5) ілюструє взаємодію між основними компонентами системи під час цього процесу.

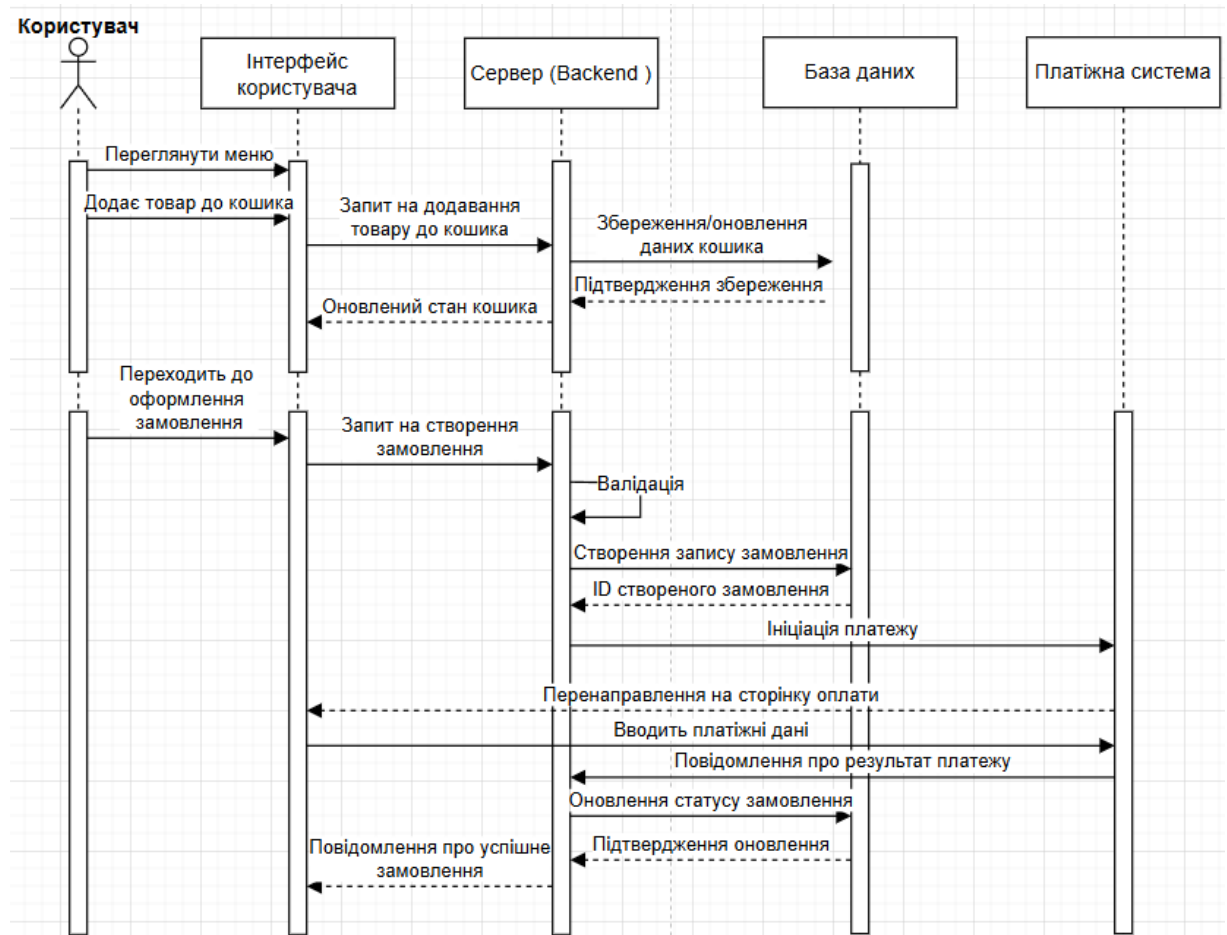


Рисунок 2.5 - Діаграма послідовності для оформлення замовлення.

Процес оформлення замовлення починається з того, що користувач обирає товари, за потреби налаштовує їх, наприклад, додаючи інгредієнти до піци, та додає до кошика. У цей момент клієнтська частина вебсайту взаємодіє з сервером, щоб оновити стан кошика в базі даних.

Після цього користувач переходить на сторінку оформлення замовлення. Якщо він не авторизований у системі, йому необхідно ввести контактні дані, такі як ПІБ, номер телефону, адресу доставки та email. Для авторизованих користувачів ці дані зазвичай підтягуються автоматично з їхнього профілю.

Далі сервер виконує перевірку коректності введених даних та наявності обраних товарів. У разі успішної валідації сервер створює новий запис у таблиці "Order" зі статусом "PENDING" (очікує оплати) і зберігає всі деталі замовлення.

На наступному етапі система перенаправляє користувача або інтегрує віджет платіжної системи для проведення оплати. Після отримання відповіді від платіжного сервісу сервер оновлює статус замовлення в базі даних на

"SUCCEEDED" у разі успіху або "CANCELLED" у разі невдачі. На завершення всього процесу користувач отримує сповіщення про фінальний результат оформлення свого замовлення.

2.4. Реалізація функціоналу вебзастосунку для піцерії

Для наочного представлення взаємодії основних програмних блоків системи розроблено діаграму компонентів (рис. 2.6). Ця діаграма є ключовим елементом у проектуванні, оскільки вона дозволяє візуалізувати високорівневу структуру вебзастосунку, чітко розмежувати його функціональні частини та зрозуміти принципи їхньої взаємодії. Діаграма демонструє, як клієнтська частина, що працює у браузері користувача, взаємодіє із серверною логікою, яка, у свою чергу, звертається до бази даних та інтегрується із зовнішніми сервісами, такими як платіжні шлюзи та поштові служби.

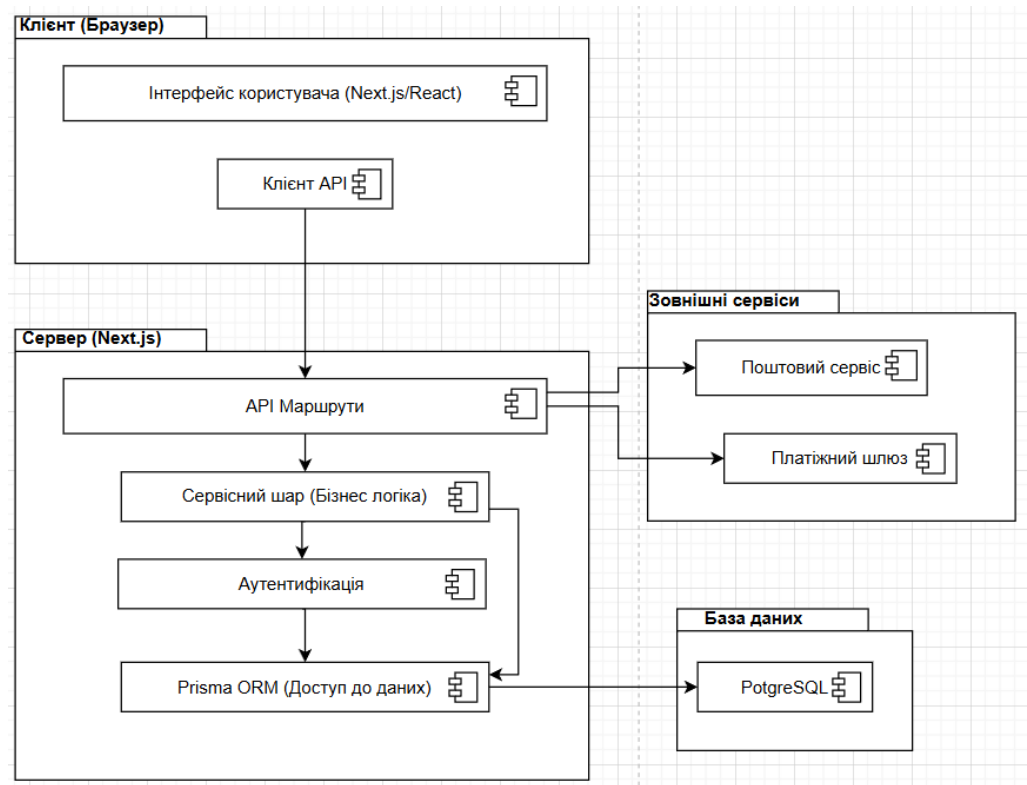


Рисунок 2.6 - Діаграма компонентів вебзастосунку.

Користувацький інтерфейс реалізовано за допомогою фреймворку Next.js та бібліотеки React [19]. Він відповідає за відображення вмісту сторінок, реагування на дії користувача і взаємодію з сервером через API-запити. У межах інтерфейсу

реалізовано сторінки каталогу товарів, кошика, оформлення замовлення, особистого кабінету та адміністративної панелі.

Взаємодія з серверною частиною здійснюється через клієнтський API, який відповідає за відправлення HTTP-запитів і обробку відповідей. Серверна логіка побудована у вигляді API-маршрутів Next.js, які обробляють запити клієнта, реалізують бізнес-логіку, звертаються до бази даних і працюють із зовнішніми сервісами. Ці маршрути охоплюють функціонал управління користувачами, товарами, категоріями, кошиком та замовленнями.

Процеси реєстрації, входу, виходу та управління сесіями користувачів реалізовані за допомогою бібліотеки NextAuth.js [14], яка також підтримує авторизацію через зовнішні провайдери (соціальні мережі).

Серце логіки системи зосереджено у сервісному шарі. Тут обробляються дані, здійснюється їх перевірка, виконуються необхідні розрахунки, як-от визначення загальної суми замовлення, та реалізується взаємодія між різними модулями застосунку.

Доступ до бази даних забезпечує Prisma ORM, яка надає типізований інтерфейс до даних PostgreSQL, дозволяючи працювати з сутностями без написання SQL-запитів вручну. Сама база даних це реляційна система PostgreSQL, яка містить усю інформацію про користувачів, товари, категорії, замовлення та інші сутності.

Обробка онлайн-платежів виконується через інтегрований платіжний шлюз, який дозволяє здійснювати безпечні фінансові транзакції. Для надсилання електронних листів, зокрема підтверджень реєстрації чи сповіщень про зміну статусу замовлення, застосовується зовнішній поштовий сервіс.

Реалізація ключових функціональних можливостей

У процесі налаштування Prisma Client було використано патерн Singleton [3] для підключення до бази даних. Такий підхід забезпечує створення лише одного екземпляра Prisma Client для всього застосунку, що підвищує ефективність з'єднання та дозволяє уникнути надмірної кількості підключень, особливо під час гарячих перезавантажень у режимі розробки. Реалізація враховує особливості

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

різних середовищ, гарантуючи коректну роботу застосунку в усіх умовах (див. додаток А).

Функціональність отримання списку категорій та товарів реалізована через відповідний API-маршрут, який також використовується на фронтенді. Зокрема, реалізовано пошук товарів за назвою з нечутливістю до регістру. Для цього застосовуються можливості фільтрації, які надає Prisma, що дозволяє повертати лише релевантні товари у форматі JSON залежно від введених користувачем даних (див. додаток Б).

Механізм реєстрації та автентифікації користувачів передбачає підтримку кількох методів входу, зокрема через електронну пошту з паролем, а також через соціальні провайдери Google та GitHub. При першому вході через соціальні мережі система автоматично створює акаунти для нових користувачів. Крім того, реалізовано збереження сесій за допомогою JWT токенів, до складу яких включено інформацію про роль користувача (див. додаток В).

Система керування кошиком підтримує роботу як для автентифікованих користувачів, так і для гостей. Ідентифікація гостей здійснюється на основі токенів. Кошик підтримує складні варіації товарів, включаючи індивідуальні комбінації інгредієнтів. Щоб уникнути дублювання, застосовується механізм групування схожих товарів. Крім того, система зберігає стан кошика між сесіями, що покращує користувацький досвід (див. додаток Г).

Процес оформлення замовлення реалізований таким чином, що вміст кошика перетворюється у повноцінне замовлення зі збереженням усіх налаштувань товарів та подальшим відстеженням статусу. Система інтегрує механізми обробки платежів, забезпечує генерацію унікальних ідентифікаторів замовлень для спрощення логістики та зберігає всі деталі, необхідні для виконання замовлення (див. додаток Д).

Для адміністраторів реалізовано функціонал управління товарами, що дозволяє створювати, редагувати та належним чином категоризувати товари. Особлива увага приділяється варіативності піци: підтримуються різні розміри, типи тіста, а також можливість додавання інгредієнтів. Також реалізовано

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

завантаження зображень, що забезпечує повноцінну візуалізацію товарів на фронтенді (див. додаток Е).

Висновки до другого розділу

На основі проведеного аналізу функціональних та нефункціональних вимог було детально розроблено структуру майбутньої системи. Важливим кроком стало визначення ключових варіантів використання, що охоплюють взаємодію різних типів користувачів із системою, зокрема гостей, зареєстрованих клієнтів та адміністраторів. Це дозволило сформулювати чітке уявлення про необхідний функціонал та логіку роботи вебзастосунку.

На наступному етапі було спроектовано об'єктно-орієнтовану структуру системи, що знайшло своє відображення у розробленій UML-діаграмі класів. Ця діаграма наочно демонструє основні сутності системи, їх атрибути та взаємозв'язки, що є фундаментом для подальшої розробки бази даних та програмної реалізації. Особливу увагу було приділено проектуванню реляційної бази даних з використанням PostgreSQL та Prisma ORM, що забезпечить надійне зберігання та ефективне управління даними про користувачів, замовлення, товари та інші важливі аспекти діяльності піцерії.

Завершальним етапом проектування стала розробка алгоритмів роботи вебзастосунку, що включає опис логіки взаємодії користувача з системою та обробки даних на серверній стороні. Було представлено діаграми активності та послідовності, які візуалізують ключові бізнес-процеси, такі як оформлення замовлення, та взаємодію між різними компонентами системи. Реалізація функціоналу вебзастосунку спирається на використання сучасних технологій, таких як Next.js для розробки користувацького інтерфейсу та серверної логіки, React для побудови динамічних компонентів, та NextAuth.js для забезпечення безпечної автентифікації та авторизації користувачів. Таким чином, у другому розділі було закладено міцний теоретичний та проєктний фундамент для подальшої розробки та впровадження функціонального вебзастосунку для піцерії.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		37

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБЗАСТОСУНКОМ ДЛЯ ПІЦЕРІЇ

3.1. Порядок встановлення та налаштування параметрів вебзастосунку

Для забезпечення повноцінної взаємодії з розробленим вебзастосунком для піцерії, система користувача повинна відповідати певним мінімальним технічним вимогам. Виконання вебсайту можливе на пристроях, що мають щонайменше 2 ГБ оперативної пам'яті та процесор з тактовою частотою не нижче 1.4 ГГц. Необхідною умовою є стабільний доступ до мережі Інтернет зі швидкістю завантаження не менше 5 Мбіт/с для забезпечення комфортної роботи з динамічним контентом та зображеннями продукції. Наявність сучасного веббраузера є критично важливою, зокрема рекомендуються останні версії Google Chrome, Mozilla Firefox, Microsoft Edge або Safari. Застарілі версії браузерів можуть не підтримувати використані в додатку сучасні веб-технології та стандарти безпеки.

Вебзастосунок функціонує як повноцінне хмарне рішення, що забезпечує миттєвий доступ до всіх функціональних можливостей через стандартний веббраузер без додаткових налаштувань або конфігурацій. Така архітектура гарантує автоматичне отримання оновлень та виправлень безпеки без втручання користувача.

Технічна інфраструктура системи характеризується високою надійністю та масштабованістю завдяки використанню провідних хмарних технологій. Платформа Vercel, обрана як основне середовище розгортання, забезпечує глобальну мережу доставки контенту, що гарантує швидкий відгук системи незалежно від географічного розташування користувача.

Для візуалізації фізичної архітектури системи та взаємодії її компонентів було розроблено діаграму розгортання. Ця діаграма демонструє, як програмні компоненти розміщуються на апаратних вузлах, ілюструючи інфраструктуру, на

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

якій функціонує додаток. На рисунку 3.1 представлено діаграму розгортання для вебзастосунку піцерії.

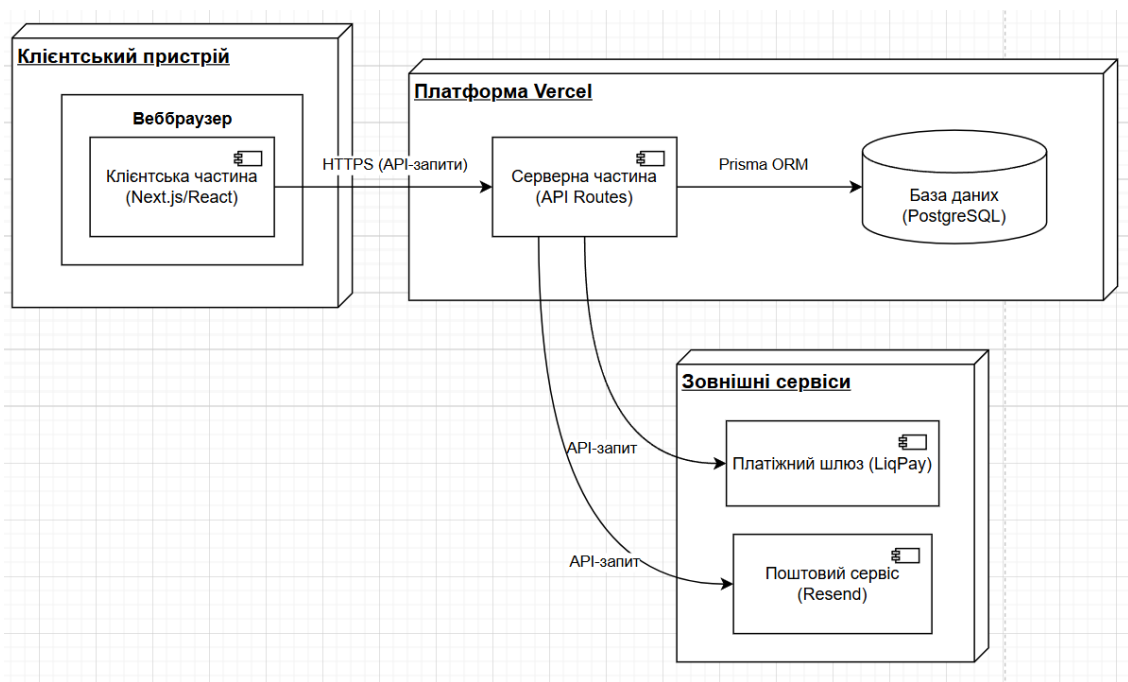


Рисунок 3.1 - Діаграма розгортання вебзастосунку.

Клієнтський пристрій, що може бути настільним комп'ютером або ноутбуком, виступає як точка взаємодії користувача з системою. На ньому за допомогою веббраузера відображається клієнтська частина застосунку, реалізована на Next.js та React. Цей інтерфейс відповідає за візуалізацію даних, обробку дій користувача та надсилання асинхронних запитів до серверної логіки через захищене HTTPS-з'єднання.

Клієнтський пристрій є комп'ютером користувача, на якому через веббраузер відбувається взаємодія з інтерфейсом застосунку. Ця клієнтська частина, реалізована на Next.js та React, надсилає запити до серверної логіки.

Уся серверна інфраструктура розгорнута на платформі Vercel. Цей вузол включає два основні компоненти. По-перше, це серверна частина, що працює на базі API-маршрутів Next.js і відповідає за обробку бізнес-логіки, автентифікацію та взаємодію з іншими частинами системи. По-друге, тут розміщено реляційну базу даних PostgreSQL, яка зберігає всю інформацію про користувачів, товари та замовлення. Взаємодія між серверною частиною та базою даних здійснюється за допомогою Prisma ORM.

Вебзастосунок також інтегровано із зовнішніми сервісами. До них належить платіжний шлюз, що забезпечує безпечну обробку фінансових транзакцій, та поштовий сервіс, який використовується для надсилання сповіщень користувачам, наприклад, для підтвердження реєстрації або зміни статусу замовлення.

3.2. Структура інтерфейсу вебзастосунку для піцерії

При першому відвідуванні сайту користувач потрапляє на головну сторінку, яка слугує вітриною закладу (Рисунок 3.2).

Тут представлено каталог продукції, згрупований за категоріями, такими як "Піци", "Сніданок", "Коктейлі" тощо. Для зручності навігації реалізовано рядок пошуку та панель фільтрації, що дозволяє відбирати товари за типом тіста, розміром чи ціною (Рисунок 3.3), тощо. У верхній частині сторінки розташовані кнопки для входу в особистий кабінет та переходу до кошика.

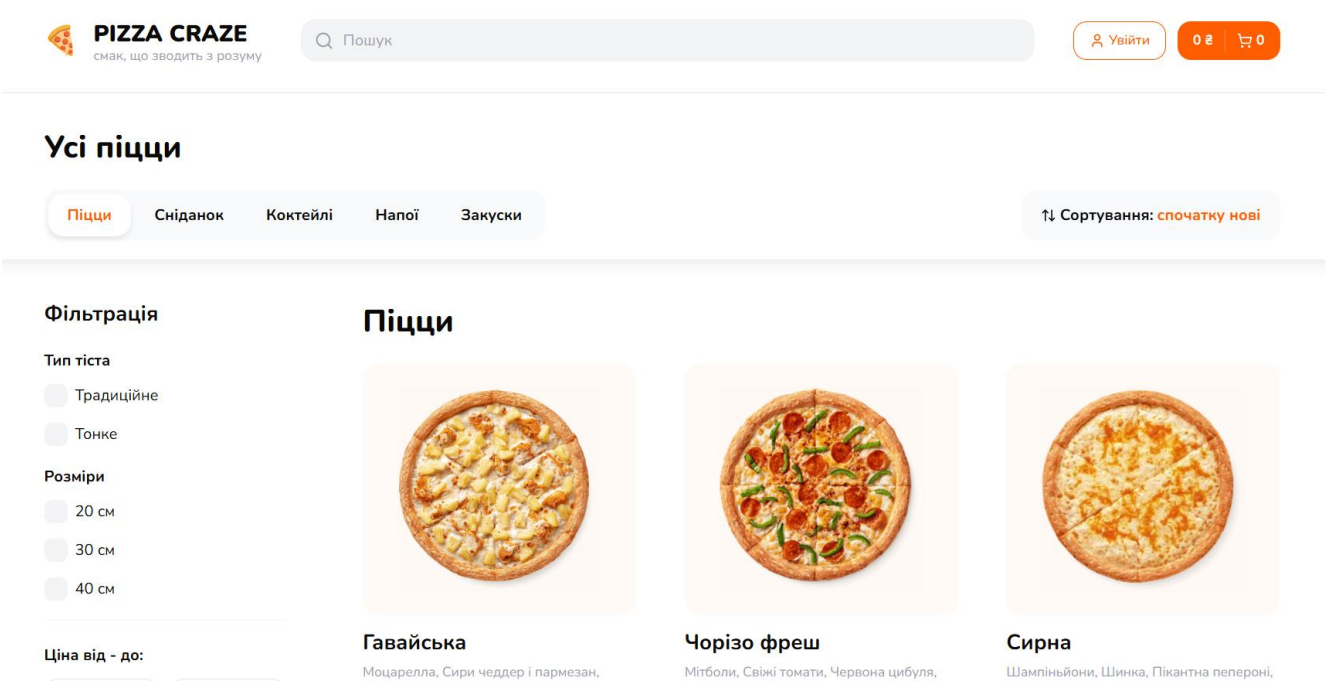


Рисунок 3.2 – Головна сторінка вебзастосунку.

Піцци

Сніданок

Коктейлі

Напої

Закуси

↑↓ Сортуння: за зростанням ціни

Фільтрація

Тип тіста

Традиційне

Тонке

Розміри

20 см

30 см

40 см

Ціна від - до:

0

160

0

160

Інгредієнти

Мітболи

Піцци

Сирна

Шампінйони, Шинка, Пікантна пепероні, Гостра чорізо, Мариновані огірочки

від 155 грн

+ Додати

Чорізо фреш

Мітболи, Свіжі помідори, Червона цибуля, Соковиті ананаси, Італійські трави, Солодкий перець, Кубики бринзи

від 159 грн

+ Додати

Сніданок

Рисунок 3.3 – Приклад сортування та фільтрації товарів.

Для отримання доступу до розширених можливостей, таких як відстеження замовлень та збереження особистих даних, користувач може створити обліковий запис. Процес реєстрації є простим та вимагає заповнення стандартної форми, що включає поля для електронної пошти, повного імені та пароля (Рисунок 3.4). Також передбачена можливість швидкої реєстрації через соціальні провайдери Google та GitHub.

Після заповнення форми на вказану електронну адресу надсилається лист із кодом для підтвердження реєстрації, що є важливим кроком для верифікації акаунта та забезпечення безпеки (Рисунок 3.5).

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

E-Mail *

vt211_zns@student.ztu.edu.ua

Повне ім'я *

Nikita Zemlianskyi

Пароль *

.....

Підтвердіть пароль *

.....

Зареєструватись

GitHub Google

Вхід

Рисунок 3.4 – Форма реєстрації користувача.

Next Pizza / Підтвердження реєстрації Зовнішній користувач Вхідні x

onboarding@resend.dev кому мені

07:04 (0 хвилин тому)

👋 Ласкаво просимо до Pizza Craze!

Дякуємо за реєстрацію! Будь ласка, використайте код нижче, щоб підтвердити вашу електронну адресу:

543988

Або просто натисніть кнопку нижче:

Підтвердити Реєстрацію

Якщо ви не реєструвалися на Pizza Craze, будь ласка, проігноруйте цей лист.

З найкращими побажаннями, команда Pizza Craze 🍕

Рисунок 3.5 – Лист для підтвердження реєстрації.

Під час вибору конкретного продукту, наприклад піци, з'являється модальне вікно з детальною інформацією (Рисунок 3.6). Тут користувач може персоналізувати своє замовлення: обрати розмір, тип тіста, а також додати додаткові інгредієнти за власним смаком. Після налаштування товар можна додати до кошика.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

Після того, як користувач завершив набір товарів і хоче перевірити, які саме елементи увійдуть до замовлення, вміст кошика доступний для перегляду у відповідному модальному вікні (Рисунок 3.7). Користувач бачить усі додані товари, їх кількість та загальну вартість, а також може редагувати замовлення перед його оформленням.

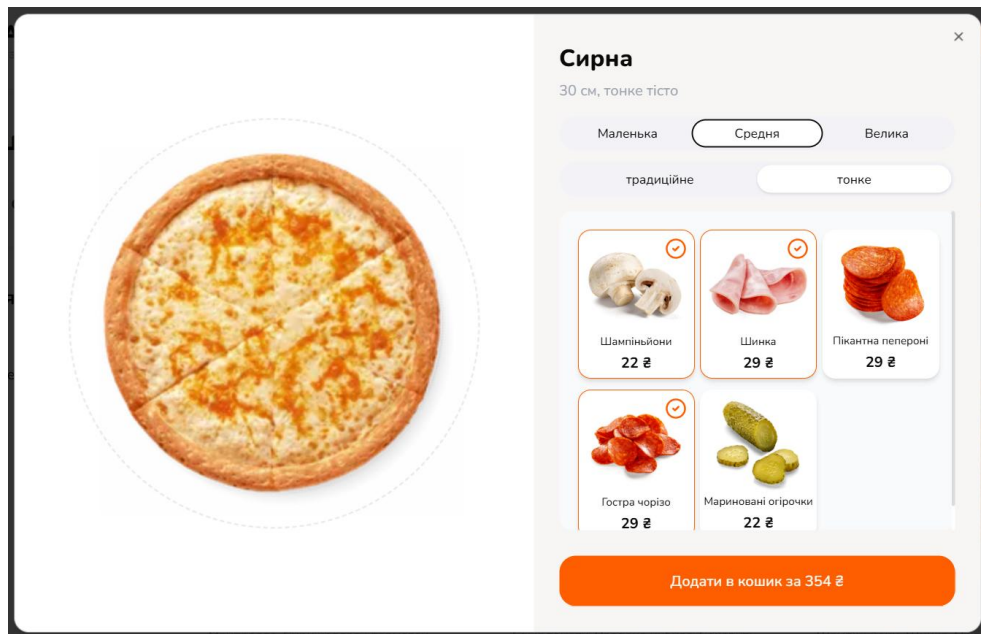


Рисунок 3.6 – Модульне вікно продукту.

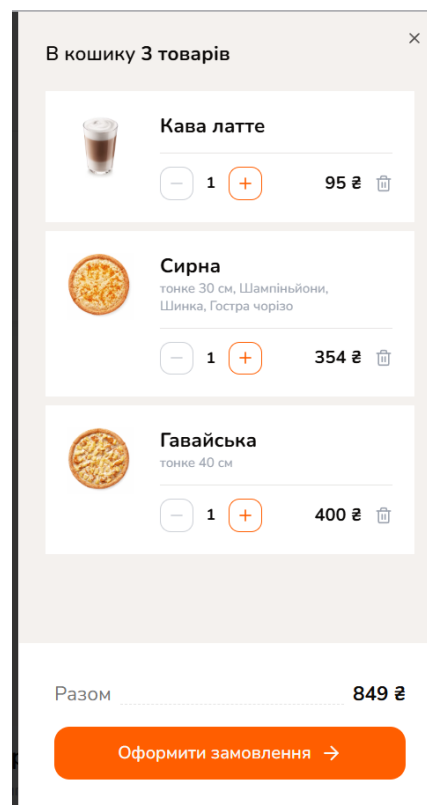


Рисунок 3.7 – Модульне вікно кошика.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

На сторінці оформлення замовлення (Рисунок 3.8) відображається фінальний список товарів, вартість доставки та загальна сума до сплати. Користувачу необхідно вказати свої контактні дані для доставки, а також обрати форму оплати. Для авторизованих клієнтів частина цих полів заповнюються автоматично.

Оформлення замовлення

1. Кошик

	Гавайська тонке 40 см	400 ₴	- 1 + ×
	Сирна тонке 30 см, Шампінйони, Шинка, Гостра чорізо	354 ₴	- 1 + ×
	Кава латте	95 ₴	- 1 + ×

Разом:

889 ₴

Вартість продуктів: 849 ₴

Доставка: 40 ₴

☐ Оплата при отриманні

[Перейти до сплати →](#)

2. Особисті дані

Nikita	Zemlianskyi
vt211_zns@student.ztu.edu.ua	1234567891

Рисунок 3.8 – Сторінка оформлення замовлення.

Після підтвердження замовлення система ініціює процес оплати. Коли користувач остаточно підтверджує замовлення, він автоматично перенаправляється на сторінку оплати, де може швидко завершити фінансову операцію. Додатково, на електронну адресу клієнта надсилається лист із дубльованим посиланням для доступу до платіжної сторінки (Рисунок 3.9). Такий підхід гарантує, що навіть у випадку технічних неполадок або бажання клієнта повернутися до процесу оплати в подальшому, необхідна інформація буде легко доступна.



Оплата Замовлення #64

Дякуємо за ваше замовлення! Будь ласка, сплатіть **889 ₴** для його завершення.

Натисніть кнопку нижче, щоб перейти до сторінки оплати:

Сплатити Замовлення

Якщо у вас виникли питання, будь ласка, зв'яжіться з нами.

З любов'ю, команда Pizza Craze ❤️

Рисунок 3.9 – Лист для сплати замовлення.

Застосунок використовує зовнішній платіжний шлюз LiqPay [9] (Рисунок 3.10), що гарантує високий рівень безпеки під час обробки фінансових операцій. Завдяки цій інтеграції, система дотримується високих стандартів шифрування та захисту даних, що значно знижує ризик несанкціонованого доступу до конфіденційної інформації. Користувач має можливість обирати найбільш зручний для себе спосіб оплати: це може бути оплата карткою, через систему Privat24 або за допомогою Google Pay.

Тестовий режим

QR-код для оплати

Використовуйте Privat24

LIQPAY

Дані про оплату

Оплата замовлення №64

До сплати: **889.00 UAH**

24 Pay

Google Pay | VISA **** 3205

або

Картка

Privat24

Рахунок

Номер картки

0000 0000 0000 0000

Термін дії

MM/YY

CVV2

...

☐ Відправити квитанцію на e-mail

Натискаючи на кнопку «Оплатити», ви підтверджуєте що

Рисунок 3.10 – Сторінка LiqPay для сплати замовлення.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

У разі успішного проведення транзакції платіжна система відображає відповідне повідомлення та надає можливість завантажити квитанцію (Рисунок 3.11).

Тестовий режим

Дякуємо!

Платіж успішно відправлено.

ID:

2658208741

Дата:

09.06.2025 07:08:56

Сума:

889.00 UAH

E-mail для отримання квитанції

Введіть e-mail

Надіслати квитанцію

Завантажити квитанцію

Рисунок 3.11 – Успішна оплата.

Одночасно на пошту користувача надходить лист, що підтверджує успішну оплату та містить деталізований список придбаних товарів (Рисунок 3.12).

onboarding@resend.dev

кому мені ▼

07:09 (0 хвилин тому)

Дякуємо за покупку! 🎉

Ваше замовлення #64 успішно сплачено!

Ось деталі вашого замовлення:

Список продуктів:

Гавайська

Ціна: 400 ₴ x 1 шт.

Всього: 400 ₴

Сирна

Ціна: 354 ₴ x 1 шт.

Всього: 354 ₴

Кава латте

Ціна: 95 ₴ x 1 шт.

Всього: 95 ₴

...

Рисунок 3.12 – Повідомлення на електронну пошту про успішну оплату.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

Зареєстровані користувачі мають доступ до особистого кабінету (Рисунок 3.13). Цей розділ дозволяє переглядати історію замовлень та їх поточний статус, а також редагувати персональні дані, зокрема змінювати пароль.


PIZZA CRAZE
смак, що зводить з розуму

Пошук

Профіль

0 а 0

Замовлення #64 09 червня 2025 р., в 07:08

Сплачено


Кава латте

95 ₴

1 шт.


Сирна
30 см, тонке тісто + ШампінйониШинкаГостра чорізо

354 ₴

1 шт.


Гавайська
40 см, тонке тісто

400 ₴

1 шт.

Загалом: 889 ₴

Особисті дані

E-Mail *
 vt211_zns@student.ztu.edu.ua

Повне ім'я *
 Nikita Zemlianskyi

Новий пароль *

Повторіть пароль *

Рисунок 3.13 – Особистий кабінет користувача з історією замовлень.

Для управління вебзастосунком передбачено окрему адміністративну панель. Її головна сторінка (Рисунок 3.14) надає швидкий доступ до ключових розділів управління та відображає основну статистику: кількість замовлень, продуктів, категорій, інгредієнтів та користувачів.

Панель адміністратора


Про адміністративну панель
Ласкаво просимо до адміністративної панелі Pizza Craze. Тут ви можете керувати замовленнями, продуктами та користувачами вашого сайту. Виберіть один із розділів нижче, щоб почати роботу.


Аналітика
Статистика та аналіз продажів


Замовлення 64
Управління замовленнями користувачів


Продукти 23
Управління асортиментом продуктів


Категорії 5
Управління категоріями продуктів


Інгредієнти 17
Управління інгредієнтами


Користувачі 9
Управління користувачами

Рисунок 3.14 – Головна сторінка панелі адміністратора.

У розділі управління замовленнями (Рисунок 3.15) адміністратор може переглядати детальну інформацію про кожне замовлення та змінювати його статус, наприклад, "Готується", "В дорозі", "Доставлено", тощо.

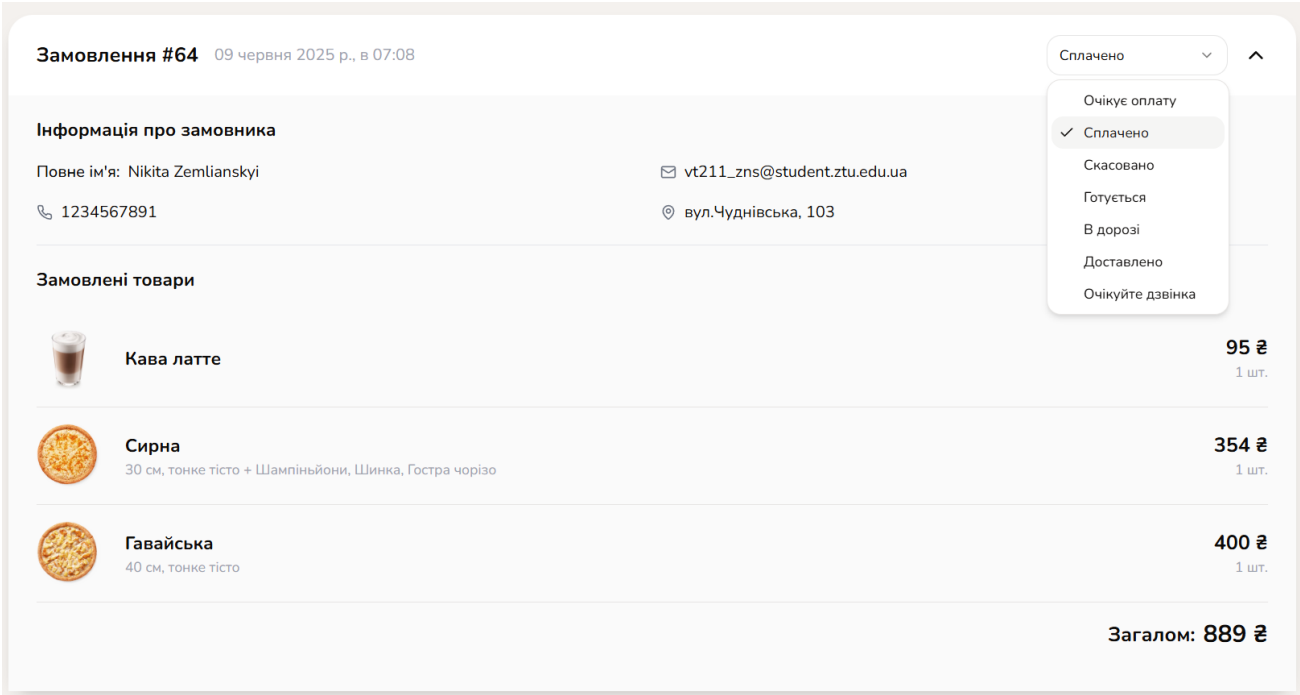


Рисунок 3.15 – Сторінка управління замовленнями.

Керування асортиментом відбувається на сторінці управління продуктами (Рисунок 3.16). Адміністратор може редагувати існуючі товари та додавати нові, заповнивши відповідну форму (Рисунок 3.17), де вказується назва, URL зображення, категорія та ціна продукту.

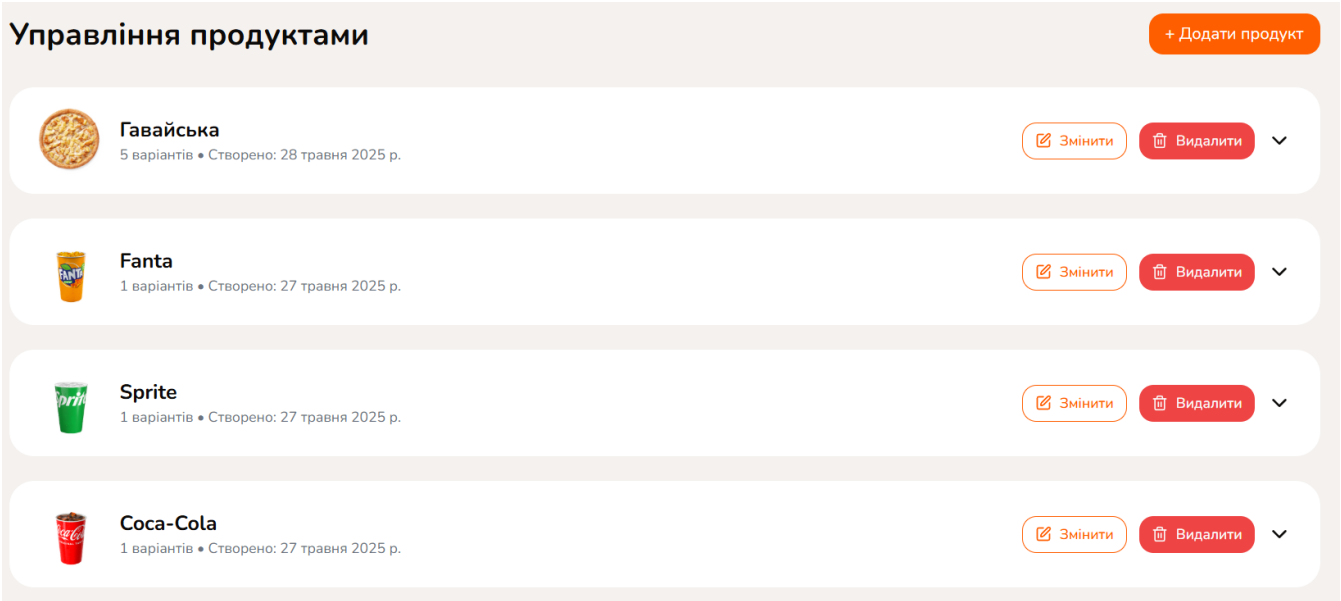


Рисунок 3.16 – Сторінка управління продуктами.

Створення нового продукту

Назва*

Введіть назву товару

URL зображення*

https://example.com/image.jpg

Категорія*

Оберіть категорію

Тип продукту*

☒ Звичайний продукт ☐ Піца

Ціна (₴)*

Ціна товару

Скасувати

Створити продукт

Рисунок 3.17 – Форма для створення нового продукту.

Важливим інструментом для аналізу бізнес-показників є сторінка аналітики (Рисунок 3.18). Тут візуалізовано дані про загальний дохід, кількість замовлень, середню вартість покупки та динаміку продажів за певний період. Окремо виведено статистику найпопулярніших продуктів, тощо.

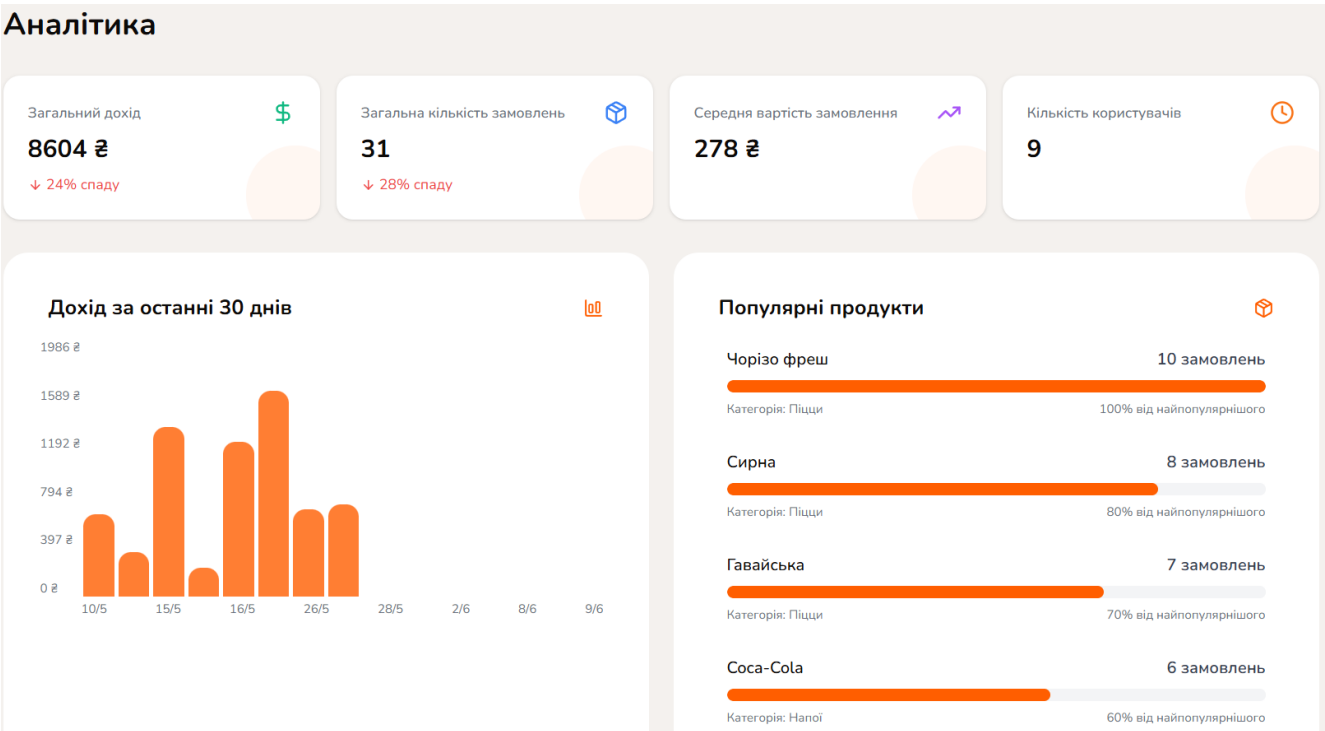


Рисунок 3.18 – Сторінка аналітики продаж.

Адміністративна панель також містить інструменти для управління інгредієнтами (Рисунок 3.19), категоріями (Рисунок 3.20) та користувачами

системи (Рисунок 3.21), що забезпечує повний контроль над усіма аспектами роботи вебзастосунку.

Управління інгредієнтами					
ID	Назва	Ціна	Зображення	Використання	Дії
14	Італійські трави	15 ₴	 Переглянути	2 продукт(ів) 0 кошник(ів)	✎ ✖
9	Гостра чорізо	29 ₴	 Переглянути	1 продукт(ів) 1 кошник(ів)	✎ ✖
4	Гострий перець халапеньйо	22 ₴	 Переглянути	2 продукт(ів) 2 кошник(ів)	✎ ✖
16	Кубики бринзи	29 ₴	 Переглянути	2 продукт(ів) 0 кошник(ів)	✎ ✖
10	Мариновані огірочки	22 ₴	 Переглянути	2 продукт(ів) 1 кошник(ів)	✎ ✖
2	Моцарелла	29 ₴	 Переглянути	2 продукт(ів) 2 кошник(ів)	✎ ✖

Рисунок 3.19 – Сторінка управління інгредієнтами.

Управління категоріями			
ID	Назва	Кількість продуктів	Дії
3	Закуси	4	✎ ✖
4	Коктейлі	4	✎ ✖
5	Напої	8	✎ ✖
1	Піцци	4	✎ ✖
2	Сніданок	3	✎ ✖

Рисунок 3.20 – Сторінка управління категоріями.

Управління користувачами						
ID	Ім'я	Email	Провайдер	Роль	Дата реєстрації	Дії
17	Nikita Zemlianskyi	vt211_zns@student.ztu.edu.ua	Email	USER	09 червня 2025 р.	✖ Видалити
14	Тестовий Користувач	userdel@gmail.com	Email	USER	28 травня 2025 р.	✖ Видалити
13	Ivan Ivanov	testPanel@gmail.com	Email	USER	28 травня 2025 р.	✖ Видалити
10	User Name	test11@gmail.com	Email	USER	15 травня 2025 р.	✖ Видалити
9	New New	test4@gmail.com	Email	USER	15 травня 2025 р.	✖ Видалити
5	New User	itali163537@gmail.com	google	USER	11 травня 2025 р.	✖ Видалити
4	Nikita	nikita163537@gmail.com	Email	USER	11 травня 2025 р.	✖ Видалити
1	New User	user@test.com	Email	USER	07 травня 2025 р.	✖ Видалити
2	Admin Test	admin@test.com	Email	ADMIN	07 травня 2025 р.	

Рисунок 3.21 – Сторінка керування користувачами.

3.3. Тестування роботи вебзастосунку

Заключним та одним із найважливіших етапів розробки програмного забезпечення є процес його всебічного тестування. Цей крок дозволяє переконатися, що всі функціональні можливості системи працюють коректно, а взаємодія користувача з інтерфейсом є зручною та інтуїтивно зрозумілою. Для перевірки працездатності розробленого вебзастосунку для піцерії було проведено серію ручних тестів, що імітують реальні сценарії використання системи різними типами користувачів.

Перший сценарій передбачав перевірку процесу реєстрації нового користувача. Перед початком тестування система перебувала у стані, де користувач не був авторизований. Було виконано перехід до форми реєстрації, заповнено поля для електронної пошти, повного імені та пароля, після чого натиснуто кнопку «Зареєструватись». Очікувалося, що система створить новий обліковий запис та надішле на вказану пошту лист для верифікації. Фактичний результат повністю збігся з очікуваним: лист із кодом підтвердження було отримано, а після його підтвердження користувач зміг успішно увійти до свого нового кабінету.

Наступний тест був спрямований на перевірку основного функціоналу – кастомізації та замовлення продукту. Будучи на головній сторінці, було обрано одну з піц, що відкрило модальне вікно з її детальним описом. Далі було змінено стандартний розмір та тип тіста, а також додано кілька додаткових інгредієнтів. Після додавання кастомізованого товару до кошика та переходу на сторінку оформлення замовлення, система коректно відобразила всі обрані параметри та перерахувала кінцеву вартість. Очікуваний результат, що полягав у правильному збереженні та відображенні всіх налаштувань товару, був повністю досягнутий.

Також було протестовано функціонал адміністративної панелі, зокрема, зміну статусу замовлення. Для цього було здійснено вхід в систему під обліковим записом адміністратора та виконано перехід до розділу управління замовленнями. У списку було знайдено тестове замовлення та його статус змінено з «Оплачено» на «В дорозі». Очікувалося, що зміна миттєво відобразиться як в адміністративній панелі, так і в особистому кабінеті клієнта. Фактичний результат підтвердив

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

коректну роботу функціоналу: статус оновився в обох інтерфейсах, що забезпечує належну комунікацію з клієнтом.

Окрім перевірки успішних сценаріїв, було проведено тестування реакції системи на некоректні дії. Зокрема, була здійснена спроба входу в систему з правильним логіном, але невірним паролем. Очікувалося, що система заблокує доступ та виведе повідомлення про помилку. Фактичний результат підтвердив наявність валідації: користувач отримав сповіщення про невірні облікові дані, а вхід до системи не відбувся.

На додаток до функціонального тестування, було проведено аналіз продуктивності вебзастосунку за допомогою сервісу Google PageSpeed Insights [4]. Цей інструмент дозволяє оцінити швидкість завантаження, доступність та загальну технічну оптимізацію сайту [26], що є критично важливими показниками для забезпечення позитивного користувацького досвіду. Результати тестування головної сторінки для настільних комп'ютерів продемонстрували високі показники (Рисунок 3.22).

Застосунок отримав оцінку 94/100 за продуктивність, 90/100 за доступність, та максимальні 100/100 за дотримання найкращих практик розробки та SEO-оптимізацію. Детальний аналіз метрик (Рисунок 3.23) показує, що ключові показники швидкості знаходяться в межах норми: перше відображення контенту (First Contentful Paint) становить 0.2 секунди, а відображення найбільшого елемента (Largest Contentful Paint) – 1.1 секунди. Такі результати свідчать про високу якість оптимізації та швидку реакцію інтерфейсу на дії користувача.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

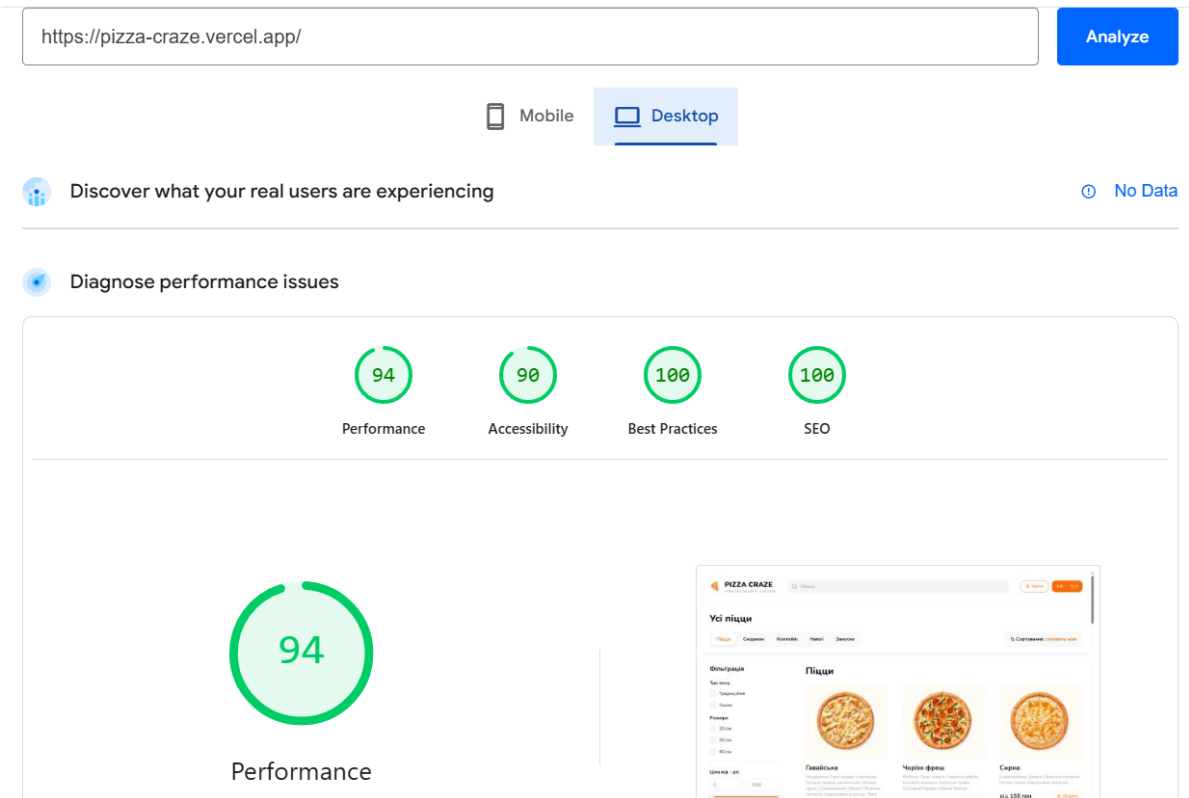


Рисунок 3.22 – Загальні результати аналізу засобами PageSpeed.

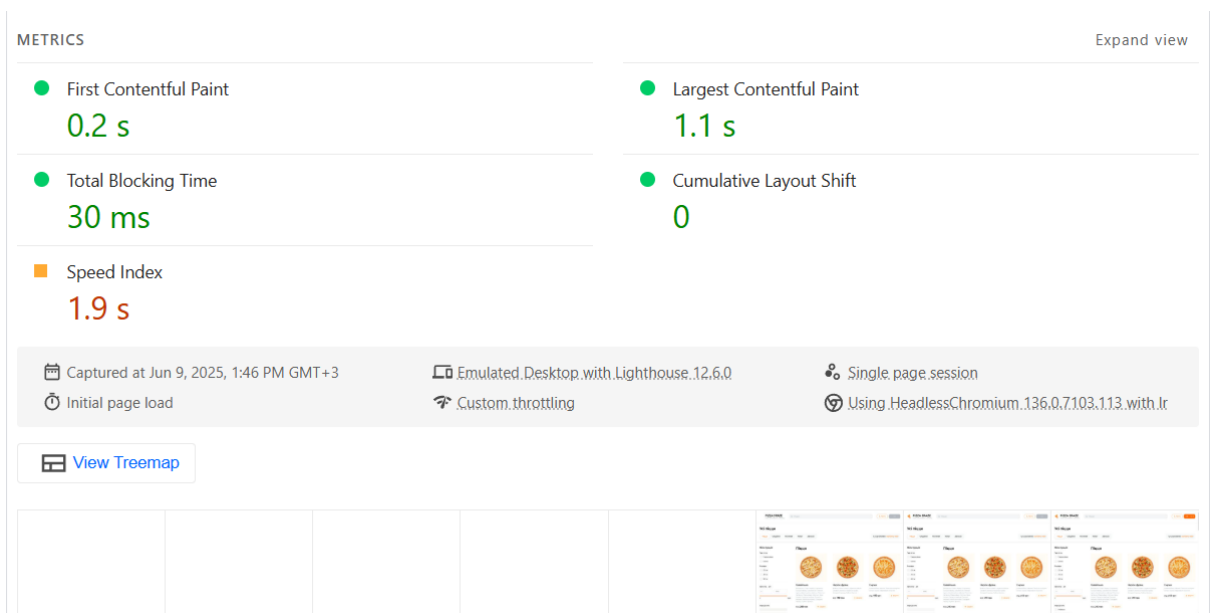


Рисунок 3.23 – Деталізовані метрики продуктивності вебзастосунку.

Висновки до третього розділу

У цьому розділі окреслено технічні вимоги для забезпечення повноцінної взаємодії користувача з вебзастосунком. Було представлено детальну діаграму розгортання, що ясно демонструє фізичну архітектуру системи та зв'язки між її

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

основними компонентами. Схема показує, що застосунок є хмарним рішенням на платформі Vercel і включає клієнтську частину, серверну логіку, базу даних PostgreSQL, а також інтеграцію із зовнішніми платіжними та поштовими сервісами.

Далі наведено повний опис структури інтерфейсу продукту. Візуально показано всі основні сторінки та функції для різних типів користувачів від перегляду каталогу товарів з фільтрами та пошуком до реєстрації, налаштування піци, оформлення замовлення і оплати через інтегровану платіжну систему. Також окреслено можливості адміністративної панелі для керування контентом, замовленнями та аналізу бізнес-показників.

Завершальним етапом розробки було тестування програмного комплексу. Ручне тестування підтвердило коректність роботи ключових функцій, а аналіз за допомогою PageSpeed продемонстрував високу швидкодію, доступність і оптимізацію продукту. Таким чином, представлений вебзастосунок повністю відповідає заданим вимогам і готовий до використання.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

ВИСНОВКИ

Під час виконання даної кваліфікаційної роботи було успішно спроектовано та розроблено програмний продукт — сучасний вебзастосунок для піцерії. Проведений на початковому етапі аналіз предметної області та наявних на ринку аналогів дозволив чітко сформулювати вимоги до майбутньої системи та визначити ключові завдання для її реалізації.

У процесі проектування було обґрунтовано вибір сучасного стеку технологій, що включає фреймворк Next.js для клієнтської частини з використанням Prisma ORM для серверної логіки. Такий вибір забезпечив високу продуктивність, гнучкість та масштабованість розробленого застосунку. Було спроектовано архітектуру системи та розроблено структуру бази даних, здатну ефективно зберігати інформацію про користувачів, замовлення та асортимент продукції зі складними варіаціями.

Основну увагу при розробці було приділено реалізації функціоналу для зручної взаємодії з системою як з боку клієнтів, так і з боку адміністрації закладу. Було створено інтуїтивно зрозумілий інтерфейс для перегляду меню та оформлення замовлень, а також реалізовано захищену адміністративну панель для керування продуктами, категоріями та відстеження замовлень.

Ключовою особливістю стала гнучка система управління товарами, що розрізняє прості продукти та комплексні, як-от піца з різними варіаціями розмірів чи тіста. Цей підхід дає змогу адміністратору гнучко налаштовувати ціни та надавати клієнтам можливість кастомізувати замовлення, додаючи інгредієнти.

У результаті було створено повноцінний та функціональний програмний продукт, що повністю відповідає поставленим цілям і готовий до впровадження. Результати роботи можуть слугувати надійною основою для подальшого розвитку проєкту, наприклад, через розробку програми лояльності або створення мобільних додатків.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

ЛІТЕРАТУРА

1. Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language User Guide – 2nd ed. – Addison-Wesley Professional, 2005 – 482 с.
2. Draw.io Documentation / diagrams.net. — 2024. Режим доступу до ресурсу: <https://www.diagrams.net/docs>.
3. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software – Addison-Wesley Professional, 1994 – 395 р.
4. Google PageSpeed Insights Documentation [Електронний ресурс] / Google Developers. - 2024. – Режим доступу до ресурсу: <https://developers.google.com/speed/docs/insights/about>.
5. Jamstack: Architectural Approach for Modern Web Applications [Електронний ресурс] / Jamstack. 2021. – Режим доступу: <https://jamstack.org>.
6. JWT.io - JSON Web Tokens Introduction. [Електронний ресурс] – Режим доступу до ресурсу: <https://jwt.io/introduction/>.
7. Kleppmann M. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. O'Reilly Media, 2017. 616 с.
8. Krug S. Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability (3rd Edition). New Riders, 2014. 216 с.
9. LiqPay API Documentation [Електронний ресурс] / LiqPay. - 2024. – Режим доступу до ресурсу: <https://www.liqpay.ua/documentation>.
10. MDN Web Docs [Електронний ресурс] / Mozilla. – 2024. – Режим доступу до ресурсу: <https://developer.mozilla.org/>.
11. Next.js 15 & PostgreSQL. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.udemy.com/course/nextjs-postgresql>.
12. Next.js Documentation: API Routes [Електронний ресурс] / Vercel. – 2024 – Режим доступу до ресурсу: <https://nextjs.org/docs/pages/building-your-application/routing/api-routes>.
13. Next.js Documentation: Getting Started [Електронний ресурс] / Vercel. – 2024 – Режим доступу до ресурсу: <https://nextjs.org/docs/getting-started>.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

- 14.NextAuth.js Documentation [Електронний ресурс] / NextAuth.js. - 2024 – Режим доступу до ресурсу: <https://next-auth.js.org/>.
- 15.OWASP Top 10: 2021 [Електронний ресурс] / Open Web Application Security Project. – 2021 Режим доступу до ресурсу: <https://owasp.org/www-project-top-ten/>.
- 16.PostgreSQL Documentation: Tutorial [Електронний ресурс] / PostgreSQL. – 2024 – Режим доступу до ресурсу: <https://www.postgresql.org/docs/current/tutorial.html>.
- 17.Prisma Documentation: Database schema [Електронний ресурс] / Prisma Data, Inc. - 2024 – Режим доступу до ресурсу: <https://www.prisma.io/docs/concepts/components/prisma-schema>.
- 18.Prisma Documentation: Quickstart Prisma Postgres [Електронний ресурс] / Prisma. – 2024 – Режим доступу до ресурсу: <https://www.prisma.io/docs/getting-started/quickstart-prismaPostgres>.
- 19.React Documentation: Main Concepts [Електронний ресурс] / Meta Platforms, Inc. 2024 – Режим доступу до ресурсу: <https://react.dev/learn>.
- 20.Shadcn UI & Next JS - Build beautiful dashboards with shadcn. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.udemy.com/course/shadcn-ui/>.
- 21.Shadcn UI Documentation: Components [Електронний ресурс] / Shadcn UI. – 2024 – Режим доступу до ресурсу: <https://ui.shadcn.com/docs/components/>.
- 22.Tailwind CSS Documentation [Електронний ресурс] / Tailwind Labs. - 2024. – Режим доступу: <https://tailwindcss.com/docs>.
- 23.The Modern React Bootcamp (Hooks, Context, NextJS, Router). [Електронний ресурс] – Режим доступу до ресурсу: <https://www.udemy.com/course/modern-react-bootcamp/>
- 24.TypeScript Documentation: Handbook [Електронний ресурс] / Microsoft Corporation. – 2024 – Режим доступу до ресурсу: <https://www.typescriptlang.org/docs/>.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

- 25.Vercel Documentation: Deployment [Електронний ресурс] / Vercel. – 2024 –
Режим доступу до ресурсу: <https://vercel.com/docs>.
- 26.Web Performance Optimization Techniques [Електронний ресурс] / Mozilla
Developer Network. - 2024. – Режим доступу до ресурсу:
<https://developer.mozilla.org/en-US/docs/Web/Performance>.
- 27.Гавerbeке М. Красномовний JavaScript. Сучасне введення у програмування.
3-тє видання. Навчальна книга - Богдан, 2021. 472 с.
- 28.Мартін Р.С. Чистий код. Створення та рефакторинг за допомогою Agile.
Фабула, 2021. 464 с.
- 29.Про захист персональних даних: Закон України від 01.06.2010 № 2297-VI //
Відомості Верховної Ради України – 2010 – № 34 – ст. 481.
- 30.Фаулер М. Рефакторинг. Поліпшення існуючого коду (2-є видання). Addison-
Wesley Professional, 2018. 448 с.

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

Налаштування Prisma Client та підключення до БД

```
import { PrismaClient } from '@prisma/client'

const prismaClientSingleton = () =>{
  return new PrismaClient()
}

declare global {
  var prismaGlobal: undefined | ReturnType<typeof prismaClientSingleton>;
}

export const prisma = global.prismaGlobal ?? prismaClientSingleton();

if (process.env.NODE_ENV !== 'production') {
  global.prismaGlobal = prisma;
}
```

Отримання списку категорій товарів

```
import { prisma } from "@prisma/prisma-client";
import { NextRequest, NextResponse } from "next/server";

export async function GET(req: NextRequest) {
  const query = req.nextUrl.searchParams.get("query") || "";

  const products = await prisma.product.findMany({
    where: {
      name: {
        contains: query,
        mode: "insensitive",
      },
    },
    take: 5,
  });

  return NextResponse.json(products);
}
```

Реєстрація користувачів та автентифікація

```
export const authOptions: AuthOptions = {
  providers: [
    GoogleProvider({
      clientId: process.env.GOOGLE_CLIENT_ID || "",
      clientSecret: process.env.GOOGLE_CLIENT_SECRET || "",
    })
  ]
}
```

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    })),
    GitHubProvider({
      clientId: process.env.GITHUB_ID || "",
      clientSecret: process.env.GITHUB_SECRET || "",
      profile(profile) {
        return {
          id: profile.id,
          name: profile.name || profile.login,
          email: profile.email,
          image: profile.avatar_url,
          role: "USER" as UserRole,
        };
      },
    })),
    CredentialsProvider({
      name: "Credentials",
      credentials: {
        email: { label: "Email", type: "text" },
        password: { label: "Password", type: "password" },
      },
      async authorize(credentials) {
        const findUser = await prisma.user.findFirst({
          where: values,
        });
        const isPasswordValid = await compare(
          credentials.password,
          findUser.password
        );

        if (!isPasswordValid) {
          return null;
        }
        return {
          id: findUser.id,
          email: findUser.email,
          name: findUser.fullName,
          role: findUser.role,
        };
      },
    })),
  ],
}

```

Додаток Г

Керування кошиком

```

import { prisma } from "@prisma/prisma-client";
import { NextRequest, NextResponse } from "next/server";
import { findOrCreateCart } from "@shared/lib/find-or-create-cart";
import { CreateCartItemValues } from "@shared/services/dto/cart.dto";
import { updateCartTotalAmount } from "@shared/lib/update-cart-total-amount";

```

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

```

export async function GET(req: NextRequest) {
  try {
    const token = req.cookies.get("cartToken")?.value;

    if (!token) {
      return NextResponse.json({ totalAmount: 0, items: [] });
    }

    const userCart = await prisma.cart.findFirst({
      where: {
        OR: [
          {
            token,
          },
        ],
      },
      include: {
        items: {
          orderBy: {
            createdAt: "desc",
          },
          include: {
            productItem: {
              include: {
                product: true,
              },
            },
            ingredients: true,
          },
        },
      },
    });

    return NextResponse.json(userCart);
  } catch (error) {
    console.log("[CART_GET] Server error", error);
    return NextResponse.json(
      { message: "Не вдалось отримати кошик" },
      { status: 500 }
    );
  }
}

export async function POST(req: NextRequest) {
  try {
    let token = req.cookies.get("cartToken")?.value;

    if (!token) {
      token = crypto.randomUUID();
    }
  }
}

```

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

```

}

const userCart = await findOrCreateCart(token);

const data = (await req.json()) as CreateCartItemValues;

// Отримати всі елементи кошика з таким самим товаром
const cartItems = await prisma.cartItem.findMany({
  where: {
    cartId: userCart.id,
    productItemId: data.productItemId,
  },
  include: {
    ingredients: true,
  },
});

// Знайти елемент кошика з точно такими самими інгредієнтами (та сама кількість і ті самі ID)
const findCartItem = cartItems.find(item => {
  // Якщо ingredients має значення undefined, обробляйте його як порожній масив
  const ingredientsToAdd = data.ingredients || [];

  // Якщо кількість інгредієнтів не збігається, це не той самий товар
  if (item.ingredients.length !== ingredientsToAdd.length) {
    return false;
  }

  // Якщо інгредієнти відсутні, відповідність визначається лише за товаром
  if (ingredientsToAdd.length === 0) {
    return true;
  }

  // Перевіряє чи всі інгредієнти у data.ingredients присутні в елементі кошика
  const ingredientIds = item.ingredients.map(ingredient => ingredient.id);
  return ingredientsToAdd.every(id => ingredientIds.includes(id));
});

// Якщо товар був знайдений, то додаємо 1
if (findCartItem) {
  await prisma.cartItem.update({
    where: {
      id: findCartItem.id,
    },
    data: {
      quantity: findCartItem.quantity + 1,
    },
  });
} else {
  await prisma.cartItem.create({

```

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    data: {
      cartId: userCart.id,
      productItemId: data.productItemId,
      quantity: 1,
      ingredients: {
        connect: data.ingredients?.map((id) => ({
          id,
        })),
      },
    },
  });
}

const updatedUserCart = await updateCartTotalAmount(token);

const resp = NextResponse.json(updatedUserCart);
resp.cookies.set("cartToken", token);
return resp;
} catch (error) {
  console.log("[CART_POST] Server error", error);
  return NextResponse.json(
    { message: "Не вдалось створити кошик" },
    { status: 500 }
  );
}
}

```

Додаток Д

Створення замовлення

```

export async function createOrder(data: CheckoutFormValues): Promise<string> {
  const cookieStore = await cookies();
  const cartToken = cookieStore.get("cartToken")?.value;
  if (!cartToken) throw new Error("Cart token not found");

  // Знаходимо корзину по токену
  const userCart = await prisma.cart.findFirst({
    include: {
      user: true,
      items: {
        include: {
          ingredients: true,
          productItem: { include: { product: true } },
        },
      },
    },
    where: { token: cartToken },
  });

  // Якщо корзина не знайдена/пуста, то викидаємо помилку
  if (!userCart || userCart.totalAmount === 0)

```

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    throw new Error("Cart is empty or not found");

const DELIVERY_PRICE = 40;

// Створюємо замовлення
const order = await prisma.order.create({
  data: {
    token: cartToken,
    fullName: `${data.firstName} ${data.lastName}`,
    email: data.email,
    phone: data.phone,
    address: data.address,
    comment: data.comment,
    totalAmount: userCart.totalAmount + DELIVERY_PRICE,
    status: data.payOnReceipt ? OrderStatus.WAITING_FOR_CALL : OrderStatus.PENDING,
    items: JSON.stringify(userCart.items),
  },
});

// Очищення кошика
await prisma.cart.update({
  where: { id: userCart.id },
  data: { totalAmount: 0 },
});
await prisma.cartItem.deleteMany({ where: { cartId: userCart.id } });

if (data.payOnReceipt) {
  // Оплата готівкою, онлайн оплата не потрібна
  return "";
}

const paymentUrl = await createPayment({
  orderId: order.id,
  amount: order.totalAmount,
  description: `Оплата замовлення №${order.id}`,
});

// Відправляємо лист-сповіщення про сплату
await sendEmail(
  data.email,
  `Next Pizza / Сплатить замовлення №${order.id}`,
  PayOrderTemplate({
    orderId: order.id,
    totalAmount: order.totalAmount,
    paymentUrl,
  })
);

return paymentUrl;
}

```

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

Адміністративні функції (управління товарами)

```
export async function createProduct(data: CreateProductData) {
  try {
    // Базова валідація
    if (!data.name || !data.imageUrl || !data.categoryId) {
      return { success: false, error: "Відсутні обов'язкові поля" };
    }

    // Створити базовий продукт
    const product = await prisma.product.create({
      data: {
        name: data.name,
        imageUrl: data.imageUrl,
        categoryId: data.categoryId,
        // Підключити вибрані інгредієнти, якщо є
        ingredients: data.ingredients && data.ingredients.length > 0
          ? {
              connect: data.ingredients.map(id => ({ id })),
            }
          : undefined
      }
    });

    // If it's a regular product (not pizza), create a single product item Якщо це
    // звичайний продукт (не піца), створити один товар
    if (!data.isPizza && data.price) {
      await prisma.productItem.create({
        data: {
          price: data.price,
          productId: product.id,
        }
      });
    }

    // Якщо це піца, створити товар для кожної варіації
    else if (data.isPizza && data.variations && data.variations.length > 0) {
      // Створити усі варіації як товарні елементи
      await Promise.all(
        data.variations.map(variation =>
          prisma.productItem.create({
            data: {
              price: variation.price,
              size: variation.size,
              pizzaType: variation.pizzaType,
              productId: product.id,
            }
          })
        )
      );
    }
  }
}
```

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    );
} else {
    return { success: false, error: "Неправильні дані продукту" };
}

// Ревалідація сторінок, щоб відобразити зміни
revalidatePath("/admin/products");
revalidatePath("/products");

return { success: true, productId: product.id };
} catch (error) {
    console.error("Error creating product:", error);
    return {
        success: false,
        error: "Помилка при створенні продукту"
    };
}
}

```

					ІПЗ.КР.Б-121-25-ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		