

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Next.js Documentation. URL: <https://nextjs.org/docs> (дата звернення: 06.05.2024).
2. Prisma ORM Documentation. URL: <https://www.prisma.io/docs> (дата звернення: 06.05.2024)Next.js Documentation. URL: <https://nextjs.org/docs> (дата звернення: 06.05.2024).
3. Prisma ORM Documentation. URL: <https://www.prisma.io/docs> (дата звернення: 06.05.2024).
4. Vercel PostgreSQL. URL: <https://vercel.com/docs/storage/vercel-postgres> (дата звернення: 06.05.2024).
5. TailwindCSS Documentation. URL: <https://tailwindcss.com/docs/installation> (дата звернення: 06.05.2024).
6. NextAuth.js Documentation. URL: <https://next-auth.js.org/getting-started/introduction> (дата звернення: 06.05.2024).
7. Zustand State Management Documentation. URL: <https://github.com/pmndrs/zustand> (дата звернення: 06.05.2024).
8. LiqPay Documentation. URL: <https://www.liqpay.ua/doc> (дата звернення: 06.05.2024).
9. Instagram Graph API Documentation. URL: <https://developers.facebook.com/docs/instagram-platform/instagram-api-with-instagram-login> (дата звернення: 06.05.2024).
10. PostgreSQL Documentation. URL: <https://www.postgresql.org/docs/current/> (дата звернення: 06.05.2024).
11. Salunkhe G., Nagpurkar S., Kengale J. A. Boosting Productivity through Deep Learning: Strategies for Enhanced Efficiency. International Journal of Intelligent Systems and Applications in Engineering. 2024. 12(13). P. 396–406
12. Методичні вказівки до лабораторних робіт з дисципліни «Емпіричні методи програмної інженерії» для студентів усіх форм навчання першого (бакалаврського) рівня вищої освіти спеціальності 121 – Інженерія програмного

забезпечення, освітня програма Програмна інженерія / Упоряд.: І.В. Груздо, О.С. Назаров. – Електронне видання. – Харків: ХНУРЕ, 2024. – 127 с.

13. Методичні вказівки до практичних робіт з дисципліни «Емпіричні методи програмної інженерії» для студентів усіх форм навчання першого (бакалавського) рівня вищої освіти спеціальності 121 – Інженерія програмного забезпечення, освітня програма Програмна інженерія / Упоряд.: І.В. Груздо, О.С. Назаров. – Харків: ХНУРЕ, 2021. – 80 с.

14. «Системні технології» 4 (153) 2024 «System technologies», с. 47-57  
DOI: 10.34185/1562-9945-4-153-2024-06

15. Пироженко, С. С. Дослідження продуктивності методів та засобів контролю глобального стану компонентів React додатків / С. С. Пироженко, Н. С. Лесна // Наука онлайн : міжнародний електронний науковий журнал. – 2020. – №10. – С. 13. – URL: <https://nauka-online.com/publications/informatsionnye-tehnologii/2020/10/doslidzhennya-produktivnosti-metodiv-ta-zasobiv-kontrolyu-globalnogo-stanu-komponentiv-react-dodatki/>.

16. Соловей, І. В. Розробка програмної системи для ведення цифрової картки домашньої тварини / І. В. Соловей // Системні технології. – 2024. – № 4 (153). – С. 47–57. – DOI: 10.34185/1562-9945-4-153-2024-06.

17. Методичні вказівки до курсового проектування з дисципліни «Бази даних» для студентів усіх форм навчання спеціальності 121 – «Інженерія програмного забезпечення» (освітньо-професійна програма «Програмна інженерія») / упоряд.: О. О. Мазурова, М. С. Широкопетлева, Ю. Ю. Черепанова, Д. О. Колесников. – Харків : ХНУРЕ, 2020. – 52 с. – URL: <https://catalogue.nure.ua/download=237424>.

ДОДАТОК А  
Vision and scope

**Vision and Scope Document**

**for**

**“Сервіс для вирішення питань пов'язаних з замовленнями в піцерії.”**

**Version 1.0**

**Prepared by**

**Мартинов Богдан Валерійович**

**ХНУРЕ**

**27.09.2024**

## Зміст (Table of Contents)

Історія ревізій.....	48
А.1 ВИМОГИ ДО БІЗНЕСУ .....	49
А.1.1 Передумови (Background).....	49
А.1.2 Можливості для бізнесу (Business Opportunity).....	49
А.1.3 Бізнес-цілі та критерії успіху (Business Objectives and Success Criteria)...	50
А.1.4 Потреби клієнтів або ринку (Customer or Market Needs) .....	51
А.1.5 Бізнес-ризиками (Business Risks).....	52
А.2 АНАЛІЗ КОНКУРЕНТІВ .....	55
А.2.1 Основні конкуренти на ринку.....	55
А.2.2 Сильні та слабкі сторони.....	56
А.2.3 Стратегії та бізнес-моделі .....	58
А.2.4 Оцінка ринкової частки та обігу.....	59
А.2.5 Висновки з аналізу .....	61
А.3 ОБРАЗ РІШЕННЯ .....	62
А.3.1 Суження про бачення проекту (Vision Statement) .....	62
А.3.2 Основний функціонал (Major Features).....	62
А.3.3 Припущення та залежності (Assumptions and Dependencies) .....	63
А.4 ОБСЯГ І ОБМЕЖЕННЯ .....	65
А.4.1 Обсяг першого випуску (Scope of Initial Release) .....	65
А.4.2 Обсяг подальших випусків (Scope of Subsequent Release).....	65
А.4.3 Обмеження та виключення (Limitations and Exclusions).....	66
А.5 БІЗНЕС-КОНТЕКСТ .....	67
А.5.1 Профілі зацікавлених сторін (Stakeholder Profiles).....	67
А.5.2 Робоче середовище (Operating Environment).....	68

**Історія ревізій**

<b>Name</b>	<b>Date</b>	<b>Reason For Changes</b>	<b>Version</b>
Мартинов Б.В.	27.09.2024	Розробка документу	Version 1.0

## **A.1 ВИМОГИ ДО БІЗНЕСУ**

### **A.1.1 Передумови (Background)**

У сучасному світі цифрових технологій та швидкого темпу життя, споживачі все більше цінують можливість швидко та зручно замовляти їжу онлайн. З розвитком інтернет-торгівлі та популярністю мобільних пристроїв, очікування клієнтів щодо доступності та якості онлайн-сервісів значно зросли.

Традиційні методи замовлення по телефону або безпосередньо в закладі не повністю задовольняють потреби сучасних користувачів, оскільки вони потребують зручних і швидких способів оформлення замовлень, що призводить до втрати потенційних клієнтів для закладів громадського харчування.

Багато піцерій стикаються з проблемами інтеграції сучасних технологій у свій бізнес через відсутність доступних та ефективних рішень. Більшість існуючих платформ такі як Domino's Pizza, Il Molino та інші, які можуть бути дорогими, складними у використанні або не адаптованими до специфіки українського ринку та місцевих платіжних систем. Розробка Web застосунку для онлайн-магазину піцерії з інтеграцією системи замовлень і оплати спрямована на вирішення цих проблем.

Веб-сервіс надасть інтуїтивно зрозумілу платформу, яка дозволить клієнтам в реальному часі переглядати меню, налаштовувати замовлення за власними вподобаннями та здійснювати онлайн-оплату через популярні українські платіжні системи. Впровадження такого рішення покращить взаємодію між піцерією та її клієнтами, підвищить рівень застосунку та сприятиме зростанню лояльності споживачів.

### **A.1.2 Можливості для бізнесу (Business Opportunity)**

Розробка веб-застосунку для вирішення питань пов'язаних з замовленнями в піцерії відкриває значні можливості для бізнесу в сфері громадського харчування

та електронної комерції. У сучасному цифровому середовищі споживачі все більше цінують зручність і швидкість онлайн-сервісів.

Наявність власного веб-застосунку дозволить піцерії значно розширити свою аудиторію, залучаючи клієнтів, які віддають перевагу замовленням через інтернет.

Система спрямована на розширення каналів збуту, дозволяючи клієнтам замовляти піцу з будь-якого місця та в будь-який час, що потенційно збільшує обсяги продажів. Інтуїтивно зрозумілий інтерфейс і можливість персоналізації замовлень підвищують задоволеність клієнтів та стимулюють повторні покупки. Підтримка українських платіжних систем забезпечує зручність та безпеку оплати, що підвищує довіру клієнтів і спрощує процес транзакцій. Автоматизація прийому та обробки замовлень зменшує навантаження на персонал, мінімізує помилки та покращує ефективність роботи закладу.

Система також включатиме маркетингові інструменти, такі як програми лояльності, акції та спеціальні пропозиції, що сприяють залученню нових клієнтів та утриманню існуючих.

### **A.1.3 Бізнес-цілі та критерії успіху (Business Objectives and Success Criteria)**

BO1: Розробити веб-сервіс, який ефективно обробляє та відображає інформацію про меню та доступність продуктів у режимі реального часу.

BO2: Інтегрувати сервіс з українськими платіжними системами для зручної та безпечної онлайн-оплати.

BO3: Забезпечити безпеку та конфіденційність персональних даних користувачів.

BO4: Створити зручний та інтуїтивно зрозумілий інтерфейс для користувачів і адміністраторів.

BO5: Забезпечити можливість сповіщень для клієнтів через електронну пошту щодо статусу замовлень.

SC1: Досягти рівня задоволеності користувачів не менше 90% протягом перших 6 місяців після запуску.

SC2: Залучити 5,000 активних користувачів протягом перших 12 місяців роботи системи.

SC3: Зменшити середній час обробки та доставки замовлень на 30% протягом перших 6 місяців використання системи.

SC4: Забезпечити стабільну роботу застосунку з показником доступності 99.9% протягом перших 12 місяців після запуску.

Виконання цих бізнес-цілей та досягнення критеріїв успіху дозволять побудувати успішну програмний веб-сервіс який підвищить ефективність та швидкість отримання гуманітарної допомоги.

#### **A.1.4 Потреби клієнтів або ринку (Customer or Market Needs)**

У сучасному цифровому світі споживачі очікують від закладів харчування зручних та швидких способів замовлення їжі. З ростом популярності онлайн-сервісів, клієнти все частіше віддають перевагу можливості замовити улюблені страви через інтернет із доставкою або самовивозом. Відсутність онлайн-присутності може призвести до втрати значної частини ринку.

Основні потреби клієнтів включають:

- зручність та швидкість замовлення — клієнти бажають мати можливість швидко та без зусиль оформити замовлення через інтернет, не витрачаючи час на телефонні дзвінки чи відвідування закладу;
- доступ до актуальної інформації — споживачі потребують детальної інформації про меню, ціни, акції та наявність продуктів, щоб приймати обґрунтовані рішення;
- безпечна та проста оплата — клієнти очікують можливості оплачувати замовлення онлайн через зручні для них українські платіжні системи, з гарантією безпеки їхніх фінансових даних;

- персоналізація замовлень — можливість налаштовувати страви за власними вподобаннями (додавати чи виключати інгредієнти) з урахуванням смакових і дієтичних уподобань, підвищує задоволеність клієнтів.

Основні потреби ринку включають:

- розширення каналів збуту — ринок потребує інструментів, які дозволять закладам громадського харчування ефективно конкурувати, пропонуючи онлайн-сервіси замовлення та оплати;

- інтеграція з локальними платіжними системами — для успішної роботи на українському ринку важливо підтримувати популярні в країні платіжні методи, забезпечуючи зручність для клієнтів;

- покращення операційної ефективності — автоматизація процесів замовлення та оплати допомагає знизити навантаження на персонал, зменшити кількість помилок та оптимізувати роботу закладу;

- підвищення лояльності клієнтів — запровадження програм лояльності, спеціальних пропозицій та персоналізованого підходу сприяє утриманню клієнтів та залученню нових.

Розробка веб-застосунку для піцерії з функціоналом онлайн-замовлень і оплат через українські платіжні системи відповідає цим потребам, надаючи як клієнтам, так і бізнесу інструменти для покращення взаємодії та підвищення ефективності роботи.

### **A.1.5 Бізнес-ризик (Business Risks)**

Розробка та впровадження програмної системи пов'язані з певними бізнес-ризиками, які можуть вплинути на успіх проекту. Основні бізнес-ризиком включають: Розробка та впровадження веб-сервісу замовлення піци пов'язані з певними бізнес-ризиками, які можуть вплинути на успіх проекту. Основні бізнес-ризиком включають:

- технічні ризиком — розробка складної системи, що включає інтеграцію з різними сервісами (платіжні системи, соціальні мережі), може зіткнутися з

технічними проблемами; непередбачені помилки, збої у функціонуванні API або проблеми з сумісністю можуть затримати запуск системи та знизити її ефективність;

– безпекові ризики — зважаючи на важливість безпеки та конфіденційності даних користувачів, існує ризик несанкціонованого доступу до системи або витоку особистої інформації; недостатні заходи безпеки можуть призвести до втрати довіри користувачів та юридичних наслідків;

– ризики сумісності — система повинна працювати на різних платформах та пристроях (десктопи, мобільні пристрої), тому існує ризик несумісності з певними браузерами або операційними системами; це може обмежити доступність веб-сервісу для певних користувачів та знизити його популярність;

– ризики впровадження — процес впровадження нової системи може зіткнутися з опором з боку користувачів, особливо якщо вони звикли до інших методів замовлення (наприклад, телефонні дзвінки); низька адаптація користувачів до нового веб-сервісу може вплинути на його ефективність та поширення;

– ризики залежності від сторонніх сервісів — інтеграція з іншими платформами, такими як платіжна система LiqPay, сервіси аутентифікації NextAuth та соціальні мережі, підвищує залежність від їхньої стабільності та надійності; збої або зміни в політиці цих сервісів можуть вплинути на функціонування нашої системи;

– ризики масштабування — успішний запуск веб-сервісу може призвести до швидкого зростання кількості користувачів, що може викликати проблеми з масштабуванням інфраструктури; невідповідність до великого навантаження може призвести до зниження продуктивності та втрати користувачів;

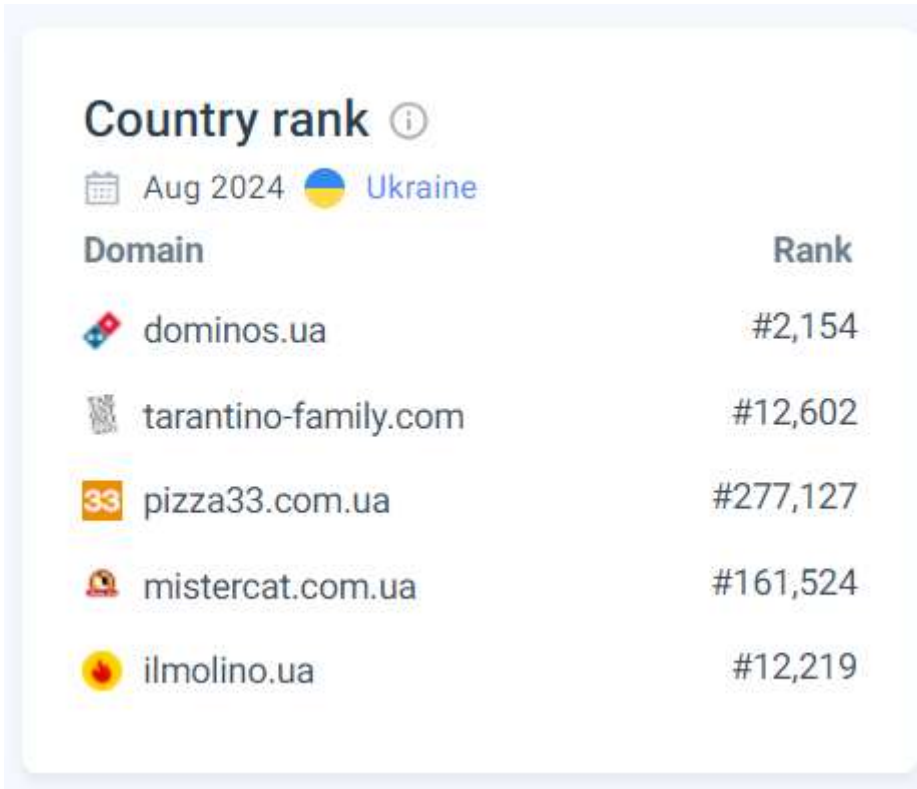
– ризики конкуренції — на ринку можуть з'явитися конкуренти з подібними або кращими рішеннями, що може знизити ринкову частку та прибутковість нашого веб-сервісу; відсутність унікальних переваг або недостатнє просування системи може призвести до її низької конкурентоспроможності.

Мінімізація цих ризиків вимагає ретельного планування, забезпечення високих стандартів безпеки, постійного моніторингу системи та готовності до швидкого реагування на непередбачені ситуації. Ефективне управління ризиками допоможе забезпечити успішне впровадження та функціонування веб-застосунку, підвищуючи його цінність для користувачів та ринку.

## A.2 АНАЛІЗ КОНКУРЕНТІВ

### A.2.1 Основні конкуренти на ринку

На ринку доставки піци в Україні діють декілька ключових гравців, які задають стандарти якості (рисунок А.1).








Country rank ⓘ	
Aug 2024 🇺🇦 Ukraine	
Domain	Rank
 dominos.ua	#2,154
 tarantino-family.com	#12,602
 pizza33.com.ua	#277,127
 mistercat.com.ua	#161,524
 ilmolino.ua	#12,219

Рисунок Б.1 – Рейтинг відвідуваності сайтів по Україні відповідно до [4].

Відповідно до рейтингу найбільш відомі, це:

#### 1. Domino's Pizza Україна[5]

Міжнародна мережа піцерій, відома швидкою доставкою та якісною продукцією. Активно розвиває онлайн-сервіси та мобільні додатки для зручності клієнтів.

#### 2. Papa John's Україна[6]:

Частина глобальної мережі, що робить акцент на використанні свіжих та якісних інгредієнтів. Пропонує широкий асортимент піц та додаткових страв.

#### 3. Pizza 33[7]:

Локальна мережа, яка фокусується на швидкій доставці піци за доступними цінами. Орієнтується на масового споживача, пропонуючи часті акції та спеціальні пропозиції.

#### 4. Mister Cat[8]:

Ресторанна мережа, що поєднує в собі доставку та атмосферу закладів. Пропонує не лише піцу, але й суші, салати та інші страви, надаючи клієнтам широкий вибір.

#### 5. Il Molino[9]:

Мережа ресторанів, яка спеціалізується на автентичній італійській піці. Використовує традиційні рецепти та високоякісні інгредієнти, орієнтуючись на преміум-сегмент ринку.

### **A.2.2 Сильні та слабкі сторони**

#### 1. Domino's Pizza Україна

Переваги:

- гарантують доставку за 30 хвилин або безкоштовно;
- інтуїтивно зрозумілий сайт та мобільний додаток;
- накопичувальні бонуси та акції для постійних клієнтів.

Недоліки:

- не всі райони міст охоплені доставкою;
- ціни вищі порівняно з конкуруючими піцеріями.

#### 2. Papa John's Україна

Переваги:

- використовують свіжі та високоякісні продукти;
- широкий вибір піц та додаткових страв;
- можливість відстежувати статус замовлення в режимі реального часу.

Недоліки:

- повільніша доставка;
- акційні пропозиції менш часті та менш вигідні.

#### 3. Pizza 33

#### Переваги:

- бюджетний варіант для широкого кола клієнтів;
- швидка доставка;
- часті акції та «комбо-набори».

#### Недоліки:

- менший вибір піц та додаткових страв;
- якість інгредієнтів нижча порівняно з іншими конкурентами.

#### 4. Mister Cat

##### Переваги:

- оригінальний стиль ресторанів та веб-сайту;
- поєднання піци, суші, салатів та інших страв;
- спеціальні пропозиції для великих замовлень.

##### Недоліки:

- веб-сайт може бути не інтуїтивно зрозумілим для користувачів.

#### 5. Il Molino

##### Переваги:

- використання традиційних італійських рецептів та технік приготування;
- високий рівень веб-застосунку та увага до деталей;
- орієнтація на клієнтів, які цінують якість понад ціну.

##### Недоліки:

- недоступні ціни для деяких категорій споживачів;
- менше знижок та спеціальних пропозицій у порівнянні з конкурентами.

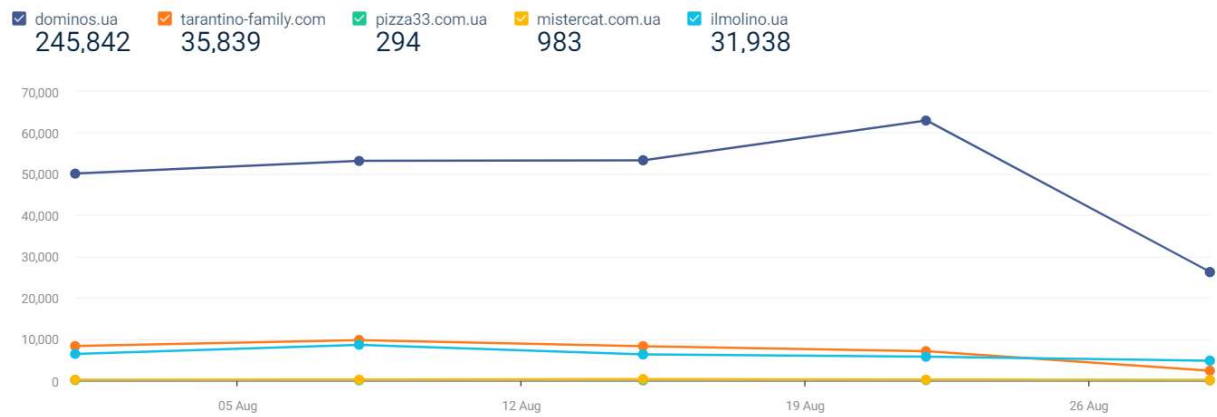


Рисунок Б.2 – Кількість користувачів по Україні за місяць.

### А.2.3 Стратегії та бізнес-моделі

#### 1. Domino's Pizza Україна:

- бізнес-модель: франчайзинг;
- стратегія: акцент на швидкості доставки та технологічних інноваціях, таких як онлайн-трекінг замовлень;
- маркетинг: програми лояльності, активна реклама в інтернеті та на телебаченні.

#### 2. Papa John's Україна:

- бізнес-модель: франчайзинг;
- стратегія: фокус на якості інгредієнтів та автентичності смаку;
- маркетинг: Підкреслення унікальних рецептів та спеціальних пропозицій для постійних клієнтів.

#### 3. Pizza 33:

- бізнес-модель: власна мережа доставки без фізичних ресторанів;
- стратегія: забезпечення доступних цін та швидкої доставки;
- маркетинг: часті акції, «комбо-набори» та спеціальні пропозиції.

#### 4. Mister Cat:

- бізнес-модель: власна мережа ресторанів;
- стратегія: поєднання ресторанного веб-застосунку з доставкою, широкий асортимент страв різних кухонь;

– маркетинг: акції для великих компаній, оригінальний дизайн та атмосфера закладів.

#### 5. Il Molino:

– бізнес-модель: власна мережа ресторанів;  
 – стратегія: орієнтація на преміум-сегмент та автентичну італійську кухню;  
 – маркетинг: підкреслення якості та автентичності, менший акцент на акціях.

### **A.2.4 Оцінка ринкової частки та обігу**

Оцінка фінансових показників приватних компаній в Україні може бути складною через обмежений доступ до офіційних даних. Проте, на основі доступної інформації з джерела [2] ([google-maps](https://www.google.com/maps)) та загальних галузевих показників можна зробити приблизні розрахунки.

#### 1. Domino's Pizza Україна:

– кількість закладів: близько 30 ресторанів у Києві;  
 – середній щоденний продаж на заклад: приблизно, 20,000 грн;  
 – річний обіг:  $30 \text{ закладів} \times 20,000 \text{ грн} \times 365 \text{ днів} \approx 219,000,000 \text{ грн}$ .

#### 2. Papa John's Україна:

– кількість закладів: близько 15 ресторанів;  
 – середній щоденний продаж на заклад: 12,000 грн;  
 – річний обіг:  $15 \times 12,000 \text{ грн} \times 365 \text{ днів} \approx 65,700,000 \text{ грн}$ .

#### 3. Pizza 33:

– кількість закладів: 15 точок доставки;  
 – середній щоденний продаж на точку: 8,000 грн;  
 – річний обіг:  $15 \times 8,000 \text{ грн} \times 365 \text{ днів} \approx 43,800,000 \text{ грн}$ .

#### 4. Mister Cat:

– кількість закладів: 25 ресторанів;  
 – середній щоденний продаж на заклад: 15,000 грн;  
 – річний обіг:  $25 \times 15,000 \text{ грн} \times 365 \text{ днів} \approx 136,875,000 \text{ грн}$ .

#### 5. Il Molino:

- кількість закладів: 10 ресторанів;
- середній щоденний продаж на заклад: 12,000 грн;
- річний обіг:  $10 \times 12,000 \text{ грн} \times 365 \text{ днів} \approx 43,800,000 \text{ грн}$ .



Рисунок Б.3 – Продажі ресторанів громадського харчування по всій Україні [2, 4]

Загальний обіг основних конкурентів: сумарно приблизно 509,175,000 грн. на рік.



Рисунок Б.4 – Продажі піцерій у порівнянні з конкурентами

Частка ринку з огляду на те, що загальний ринок піци та готових до вживання страв в Україні оцінюється приблизно в 3,591,600,000 грн, основні конкуренти займають значну, але не повну частку ринку. Точна частка кожного конкурента потребує додаткових досліджень.

### **А.2.5 Висновки з аналізу**

Ринок доставки піци в Україні є конкурентним і динамічним, з кількома великими гравцями, які пропонують подібні послуги з різними акцентами — швидкість, якість, ціна або автентичність.

#### **Стратегічні переваги проекту:**

- надання клієнтам можливості створювати власну піцу, вибираючи розмір та інгредієнти, що підвищує залученість та задоволеність споживачів;
- Забезпечення зручності та безпеки оплати, орієнтоване на локальний ринок;
- Використання сучасних технологій для покращення користувацького досвіду, включаючи інтуїтивно зрозумілий інтерфейс та інтеграцію з соціальними мережами;
- Можливість швидко реагувати на змінні потреби ринку та впроваджувати нові функції у наступних версіях продукту;
- Глибоке розуміння потреб українських споживачів та адаптація веб-застосунку відповідно до цих потреб.

Враховуючи сильні та слабкі сторони конкурентів, наш проект має потенціал виділитися за рахунок інноваційного підходу, орієнтації на персоналізацію та тісну інтеграцію з локальними сервісами. Це дозволить залучити клієнтів, які шукають нові та зручні способи замовлення улюбленої піци, і забезпечить успіх на конкурентному ринку.

## **А.3 ОБРАЗ РІШЕННЯ**

### **А.3.1 Судження про бачення проекту (Vision Statement)**

Цей проект, інноваційний онлайн-сервіс для замовлення піци та інших страв, пропонує потужне та інноваційне рішення для оптимізації процесу замовлення в піцерії. Бачення цього проекту полягає у створенні ефективної та зручної платформи, яка забезпечить швидкий і безперебійний процес онлайн-замовлення без необхідності телефонних дзвінків, з інтеграцією різних сучасних методів оплати.

Система дозволяє клієнтам легко налаштовувати свої замовлення, додаючи додаткові інгредієнти та змінюючи розмір піци в режимі реального часу.

Система пропонує інтуїтивно зрозумілу, інноваційну та легкодоступну платформу, яка дозволяє клієнтам легко реєструватися, оновлювати свої дані та керувати налаштуваннями профілю, покращуючи зручність використання веб-застосунку.

Крім того, система забезпечує точність замовлень через використання модальних вікон для перегляду деталей товару без переходу на окрему сторінку, що мінімізує помилки та покращує користувацький досвід.

Впровадження інтеграції з соціальними медіа, зокрема «Сторіс» з Instagram, підвищує залученість користувачів, дозволяючи їм ознайомлюватися з продукцією та покращує навігацію по сайту, стимулюючи соціальну взаємодію.

### **А.3.2 Основний функціонал (Major Features)**

Програмний продукт має наступний основний функціонал:

- реєстрація та авторизація користувачів;
- сортування та фільтрація страв;
- керування кошиком замовлень;
- створення власної піци;
- онлайн-замовлення;
- інтеграція з платіжною системою;

- модальні вікна для перегляду деталей товару;
- взаємодія з електронною поштою;
- перегляд статусу замовлення;
- інтеграція «Сторіс» з Instagram.

### **A.3.3 Припущення та залежності (Assumptions and Dependencies)**

Для правильної роботи програмної системи необхідно виконання наступних припущень та залежностей:

Технологічні залежності:

- використання фреймворку Next.js для розробки серверної частини та фронтенду;
- розгортання серверної частини на платформі Vercel для забезпечення стабільності та швидкодії додатку;
- використання PostgreSQL як основної бази даних у поєднанні з ORM Prisma для ефективного управління даними;
- розробка інтерфейсу користувача з використанням TailwindCSS для адаптивної та швидкої верстки;
- застосування NextAuth для аутентифікації та авторизації користувачів, з підтримкою логіну через логін-пароль та Google аккаунт;
- інтеграція з платіжною системою LiqPay для здійснення онлайн-оплат;
- використання Zustand для управління станом додатку, забезпечуючи ефективну роботу з даними;
- система контролю версій GitHub для управління кодом та спільної роботи над проектом.

Зовнішні залежності:

- стабільна робота сервісів LiqPay, NextAuth, Instagram API та інших сторонніх API, які інтегровані у веб-застосунок;
- доступність та коректна робота електронної пошти для надсилання повідомлень користувачам.

Припущення:

- користувачі мають доступ до інтернету та сучасного веб-браузера, сумісного з технологіями, використаними в проекті;
- Платформа Vercel підтримує необхідну масштабованість та продуктивність для обслуговування очікуваної кількості користувачів.

Правові залежності:

- використання відкритих джерел та технологій, які не суперечать законодавству України;
- дотримання вимог щодо захисту персональних даних та електронної комерції відповідно до чинного законодавства.

## **A.4 ОБСЯГ І ОБМЕЖЕННЯ**

### **A.4.1 Обсяг першого випуску (Scope of Initial Release)**

У початковий реліз продукту мають увійти наступні функції:

- авторизація та реєстрація з використанням електронної пошти.
- авторизація та реєстрація користувачів з використанням електронної пошти та через Google акаунт;
- перегляд каталогу страв з можливістю детального ознайомлення з кожною стравою;
- сортування та фільтрація страв за різними критеріями із збереженням налаштувань у URI;
- додавання страв до кошика, зміна кількості або видалення товарів з кошика;
- створення власної піци шляхом вибору розміру та додавання або видалення інгредієнтів;
- оформлення замовлення з вказанням адреси доставки та контактних даних;
- інтеграція з платіжною системою LiqPay для здійснення онлайн-оплати замовлень; отримання електронних листів з підтвердженням реєстрації, оформлення замовлення та після успішної оплати;
- перегляд статусу замовлення (у черзі, відхилене, в очікуванні, оплачене);
- відправлення сповіщень на електронну пошту про статус замовлення;
- впровадження «Stories» з Instagram для ознайомлення користувачів з акціями, новинками та спеціальними пропозиціями;
- адаптивний дизайн інтерфейсу для коректного відображення не тільки на комп'ютері, но і на мобільних телефонах.

### **A.4.2 Обсяг подальших випусків (Scope of Subsequent Release)**

У наступних версіях можуть бути реалізовані наступні функції:

- додавання підтримки різних мов до системи.
- додати адміністративну панель для управління каталогом страв, замовленнями та користувачами;
- розробити мобільний додаток для платформ iOS та Android для підвищення зручності користування сервісом;
- додати можливість відстеження доставки замовлення в реальному часі.

#### **A.4.3 Обмеження та виключення (Limitations and Exclusions)**

У запланованих версіях будуть наявні наступні обмеження:

- Хостинг, на якому буде знаходитись сайт буде працювати з недостатньою продуктивністю та потужністю.

## A.5 БІЗНЕС-КОНТЕКСТ

### A.5.1 Профілі зацікавлених сторін (Stakeholder Profiles)

<b>Зацікавлена сторона</b>	<b>Основна цінність</b>	<b>Позиції</b>	<b>Основні інтереси</b>	<b>Обмеження</b>
Клієнти (користувачі)	Зручний та швидкий спосіб замовлення піци онлайн	Кінцеві користувачі веб-застосунку	Швидке оформлення замовлень, широкий вибір страв, зручний інтерфейс, безпечна оплата	Технічна грамотність, доступ до інтернету та сучасного браузера
Власник піцерії	Збільшення продажів та покращення обслуговування клієнтів	Керівник бізнесу	Розширення клієнтської бази, автоматизація процесів замовлення, зменшення витрат на обробку замовлень	Бюджетні обмеження, необхідність підтримки та оновлення системи
Розробник	Реалізація якісного продукту, що задовольняє потреби користувачів та бізнесу	Відповідальний за розробку та впровадження системи	Створення функціонального та надійного додатку, використання сучасних технологій, професійний розвиток	Обмеження часу та ресурсів, технічні складнощі

<b>Зацікавлена сторона</b>	<b>Основна цінність</b>	<b>Позиції</b>	<b>Основні інтереси</b>	<b>Обмеження</b>
Платіжна система (LiqPay)	Надання послуг з безпечної онлайн-оплати	Постачальник платіжних послуг	Збільшення транзакцій через їх веб-застосунки, забезпечення безпеки та швидкості платежів	Регуляторні вимоги, технічна інтеграція, стабільність веб-застосунку
Сторонні сервіси (NextAuth, Instagram API)	Інтеграція своїх сервісів у додаток	Постачальники технологічних рішень	Підвищення використання їх сервісів, забезпечення стабільної роботи API	Можливі зміни в API, технічна підтримка, умови використання
Державні органи (регулятори)	Дотримання законодавства та нормативних актів	Регуляторні установи	Забезпечення відповідності законодавству щодо захисту даних та електронної комерції	Регуляторні вимоги, можливі зміни законодавства, необхідність ліцензування
Постачальники і інгредієнтів	Стабільне постачання якісних продуктів	Партнери по постачанню	Підтримання довгострокових відносин, збільшення обсягів продажів	Залежність від умов ринку, логістичні виклики

### **A.5.2 Робоче середовище (Operating Environment)**

Робоче середовище програмної системи замовлення піци передбачає використання сучасних технологій та платформ. Користувачі системи можуть

отримувати доступ до неї через веб-браузери на різних пристроях, таких як персональні комп'ютери, ноутбуки, смартфони та планшети.

Для роботи з програмним забезпеченням у ролі користувачів, адміністраторів та технічної підтримки необхідні наступні складові:

- операційна система — система може бути використана на різних операційних системах, таких як Windows, macOS, Linux для персональних комп'ютерів та ноутбуків, а також Android та iOS для мобільних пристроїв;

- веб-браузери — робоче середовище підтримується у всіх сучасних веб-браузерах, таких як Google Chrome, Mozilla Firefox, Microsoft Edge, Safari тощо;

- інтернет-з'єднання — для доступу до системи потрібне стабільне підключення до Інтернету, оскільки сервіс базується на веб-технологіях та використовує онлайн-сервіси для забезпечення користувачам доступу до функціоналу замовлення піци.

Для роботи з мобільним додатком (в майбутніх випусках):

- мобільні операційні системи — додаток може бути встановлений на пристроях з операційними системами Android та iOS;

- робочі середовища — мобільний додаток може бути використаний на смартфонах та планшетах, що працюють під управлінням відповідних операційних систем.

Це робоче середовище дозволяє користувачам отримувати доступ до веб-застосунку замовлення піци з різних пристроїв та операційних систем, що забезпечує зручність та доступність використання. Завдяки цьому користувачі можуть замовляти улюблені страви в будь-який час та з будь-якого місця, де є доступ до Інтернету.

## ДОДАТОК Б

### Прототип інтерфейсу користувача

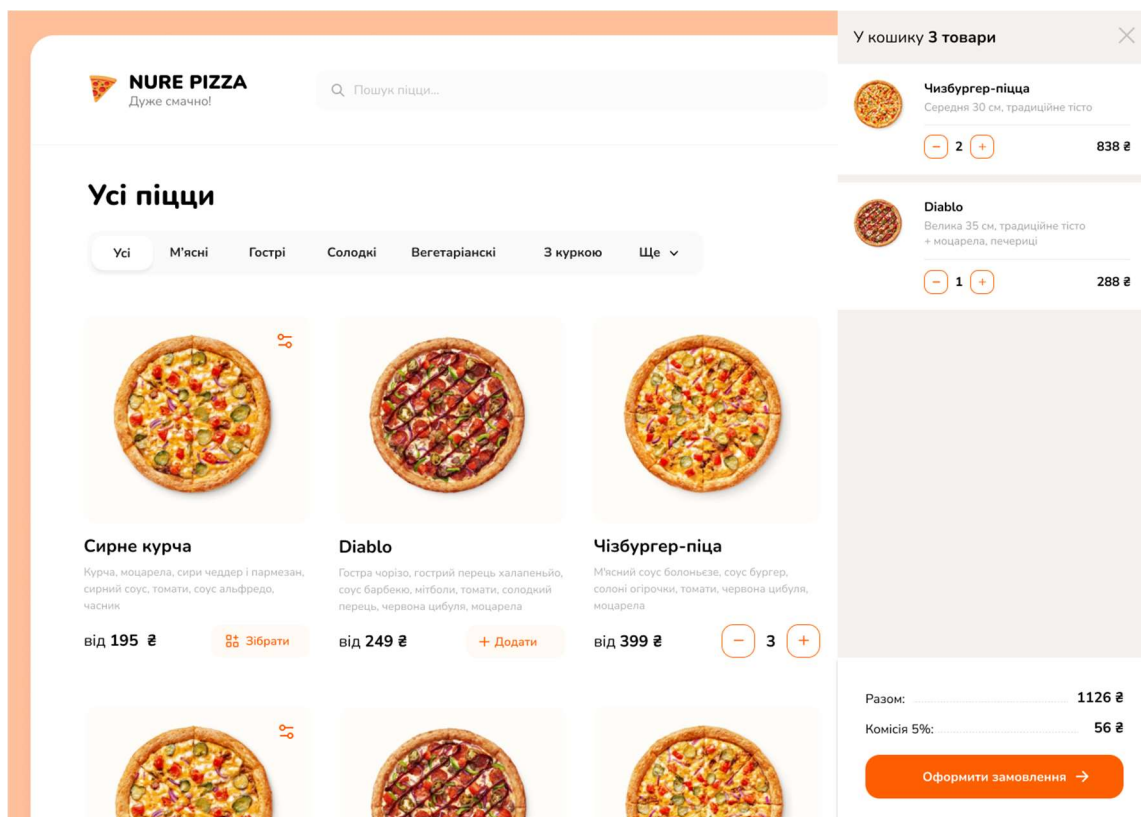


Рисунок Б.1 – Інтерфейс вибраних товарів у каталозі (рисунок виконано самостійно)

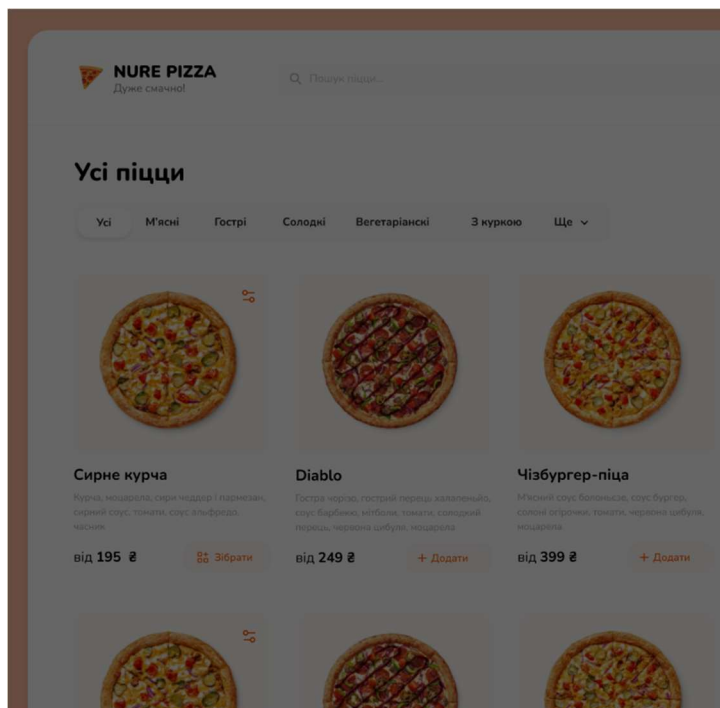


Рисунок Б.2 – Інтерфейс кошика у стані «порожній» (рисунок виконано самостійно)

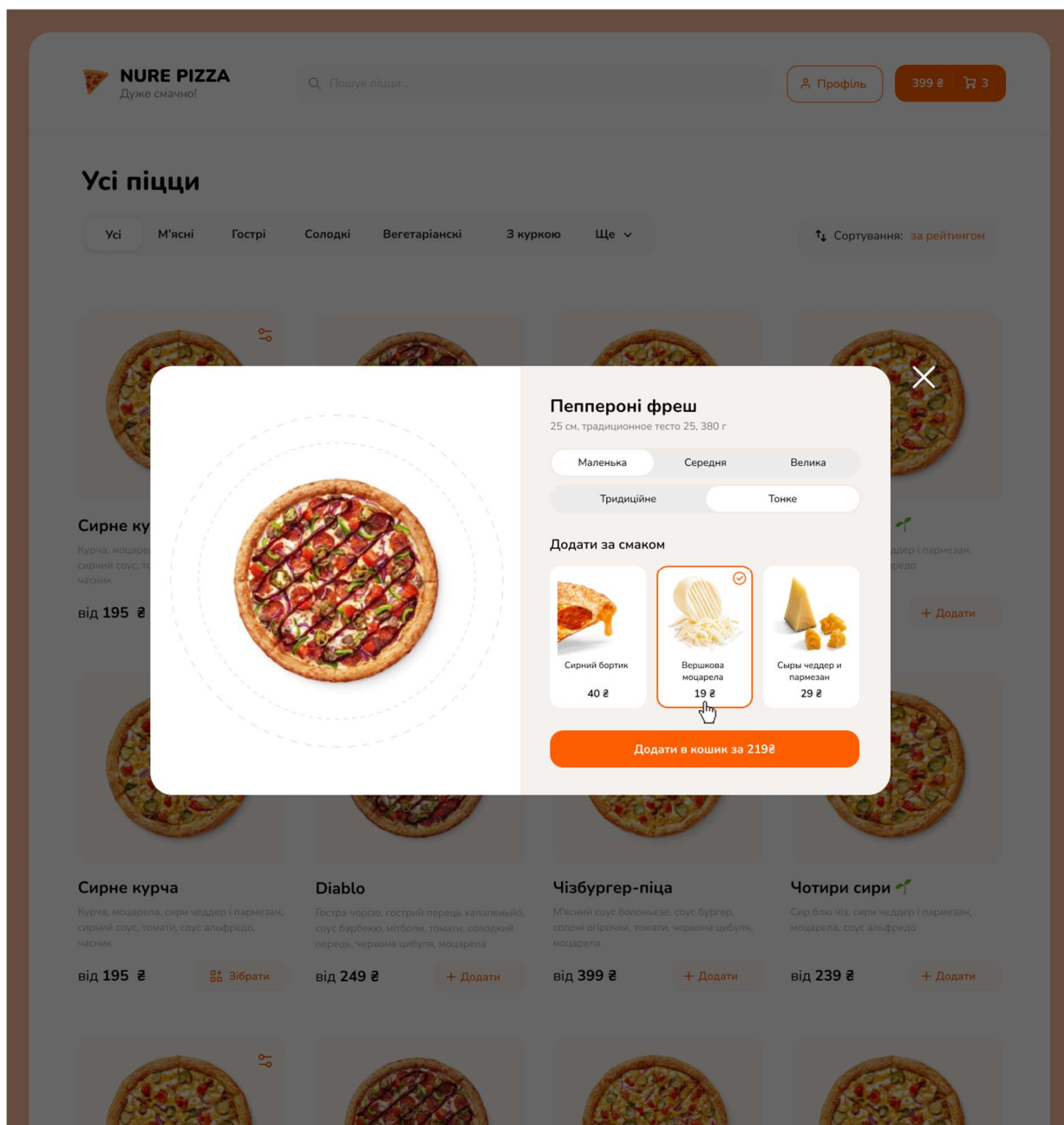




Рисунок Б.3 – Інтерфейс створення замовлення з додатковими інгредієнтами (рисунок виконано самостійно)



Профиль
399 ₴
🛒 2

Головна / Піцци / Пеппероні фреш




## Пеппероні фреш

25 см, традиционное тесто 25, 380 г


Маленька
Средня
Велика

Традиційне
Тонке


**Інгредієнти**




Сирний бортик  
40 ₴



Вершкова моцарела  
19 ₴




Сири чеддер і пармезан  
29 ₴



Ніжне курча  
39 ₴

Додати в кошик за 219₴


### Рекомендації



**Сирне курча**

Курча, моцарела, сири чеддер і пармезан, сирний соус, томати, соус альфредо, часник


від 195 ₴ + Додати



**Diablo**

Гостра чорізо, гострий перець халапеньйо, соус барбекю, мітболи, томати, солодкий перець, червона цибуля, моцарела


від 249 ₴ + Додати



**Чізбургер-піца**

М'ясний соус болоньезе, соус бургер, солоні огірочки, томати, червона цибуля, моцарела

від 399 ₴ - 1 +



**Пеппероні фреш**

Пеппероні, моцарела, томати, зелений перець, орегано, томатний соус, базилік

від 200 ₴ 🛒 Зібрати

Рисунок Б.4 – Інтерфейс сторінки детального перегляду піци з рекомендаціями (рисунок виконано самостійно) (рисунок виконано самостійно)

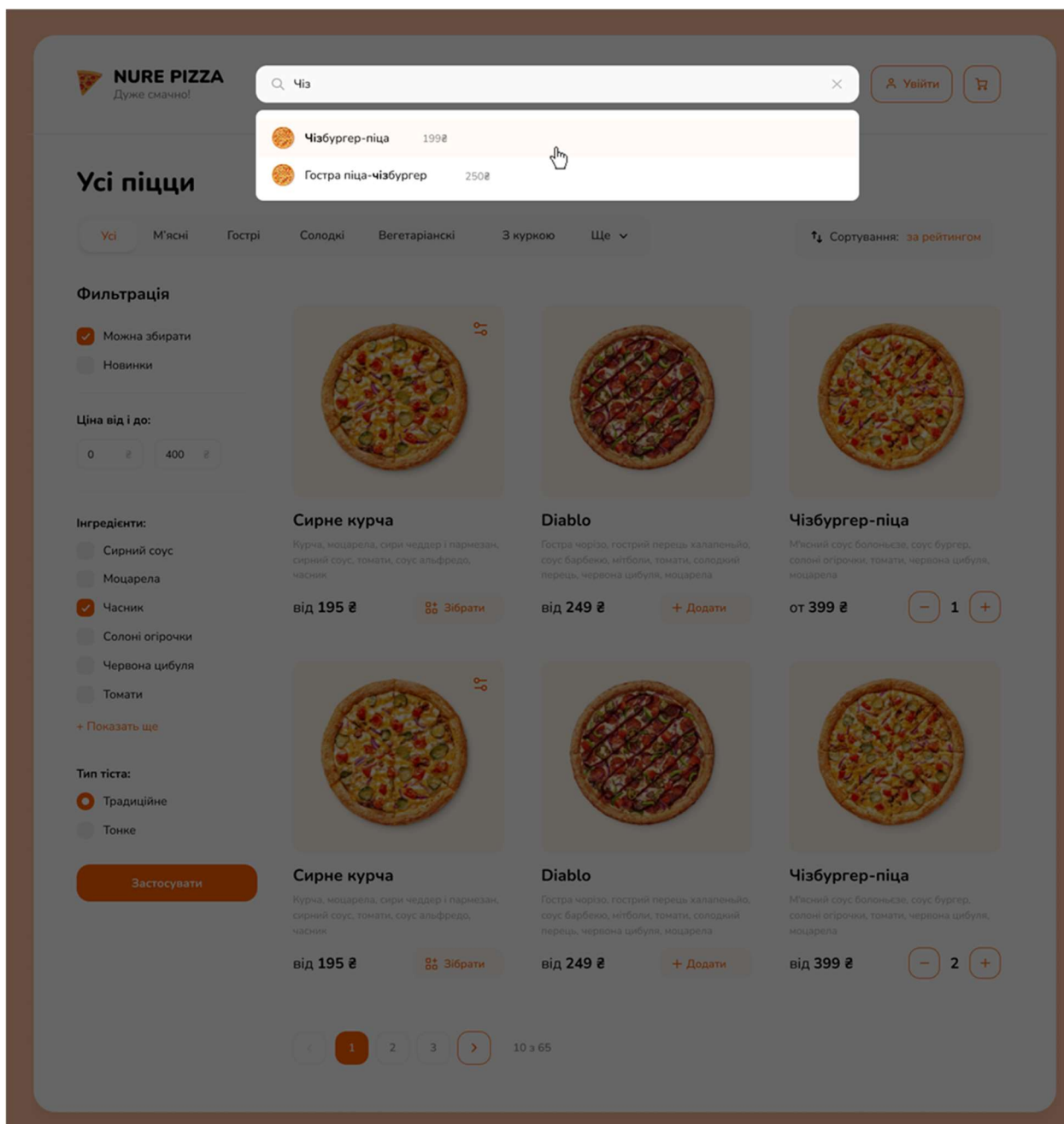


Рисунок Б.5 – Інтерфейс пошуку товару з автоматичними підказками (рисунок виконано самостійно)

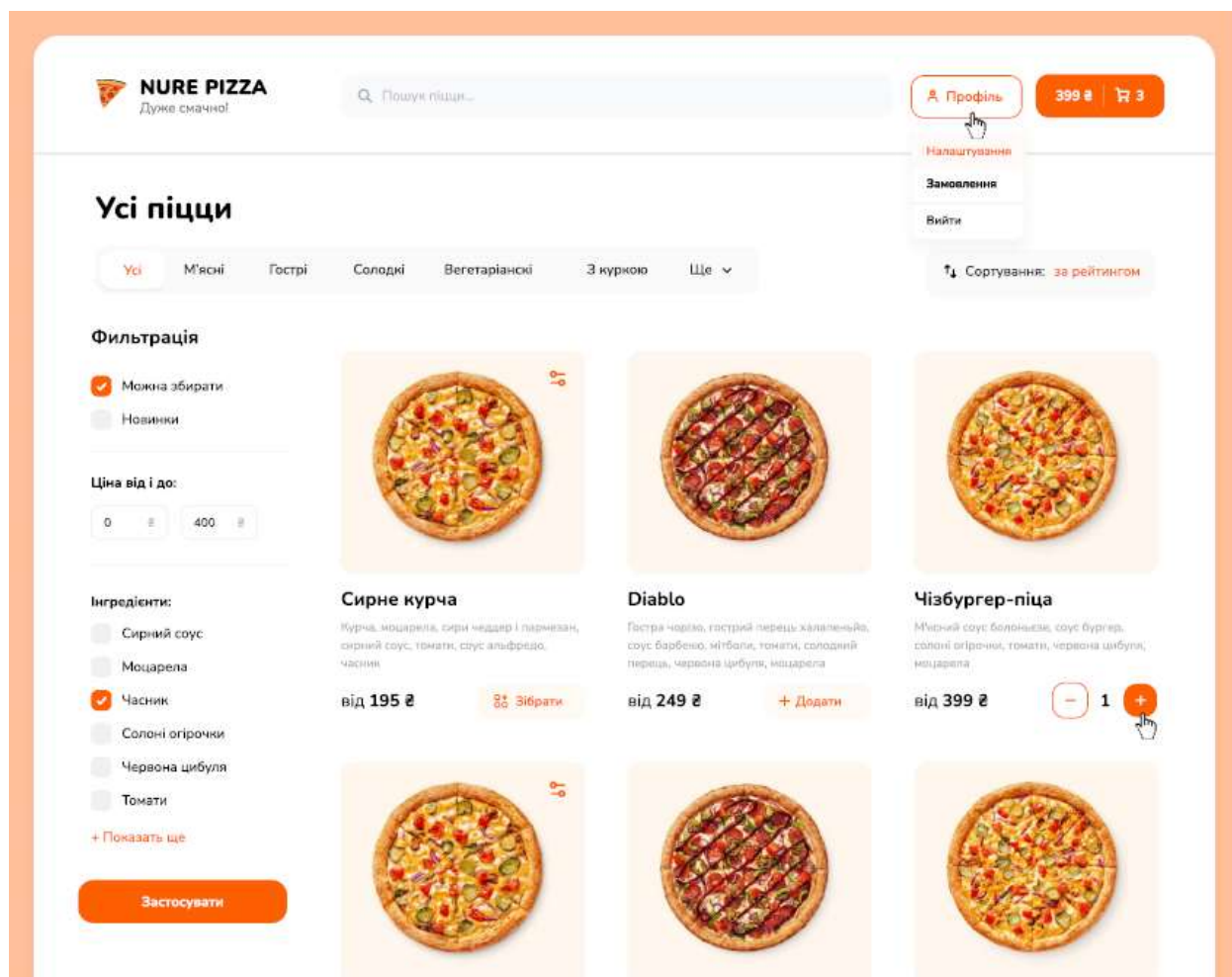


Рисунок Б.6 – Інтерфейс каталогу піц з фільтрацією та відкритим меню профілю користувача (рисунок виконано самостійно)

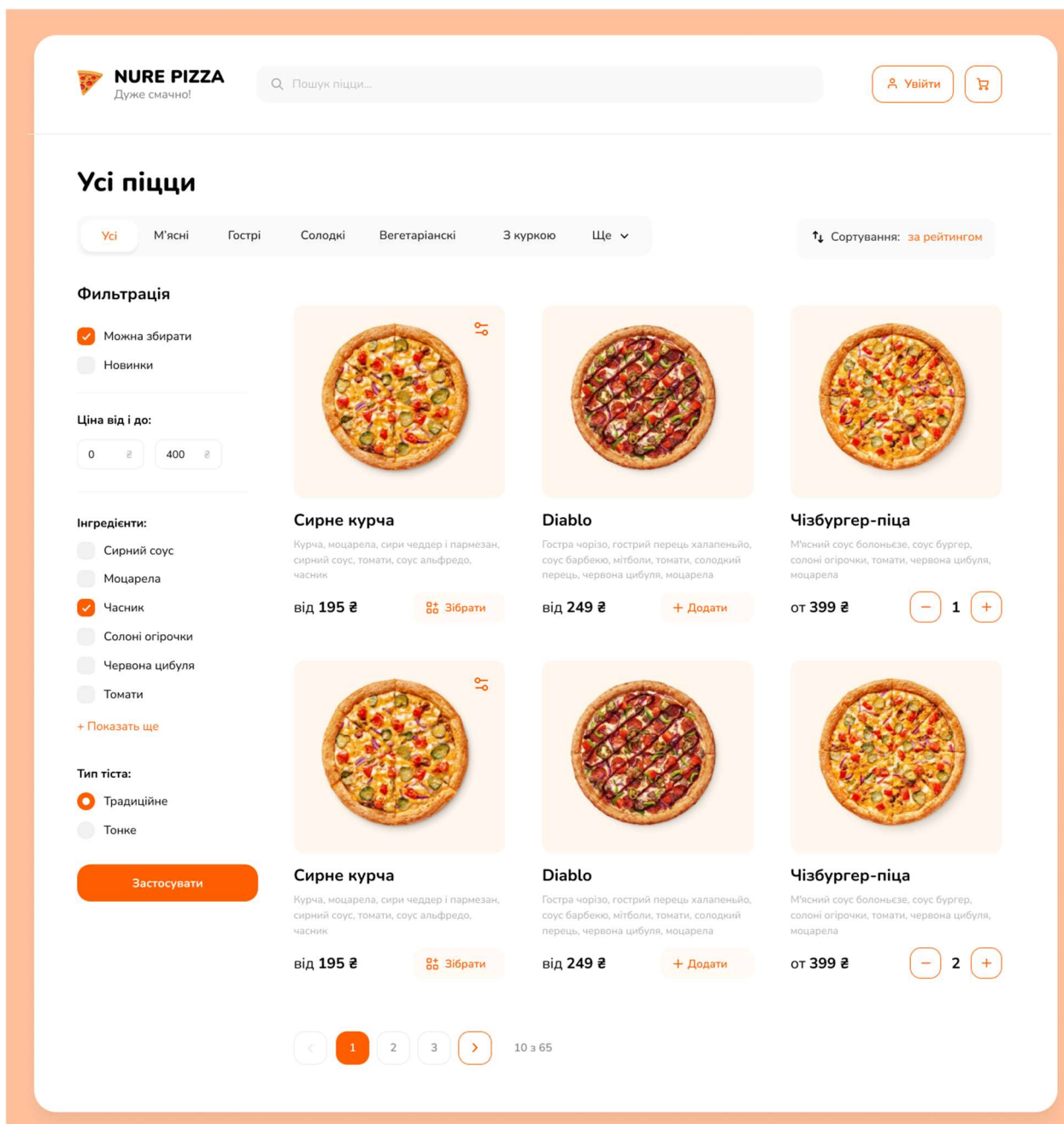


Рисунок Б.7 – Інтерфейс каталогу піц, головна сторінка(рисунок виконано самостійно)

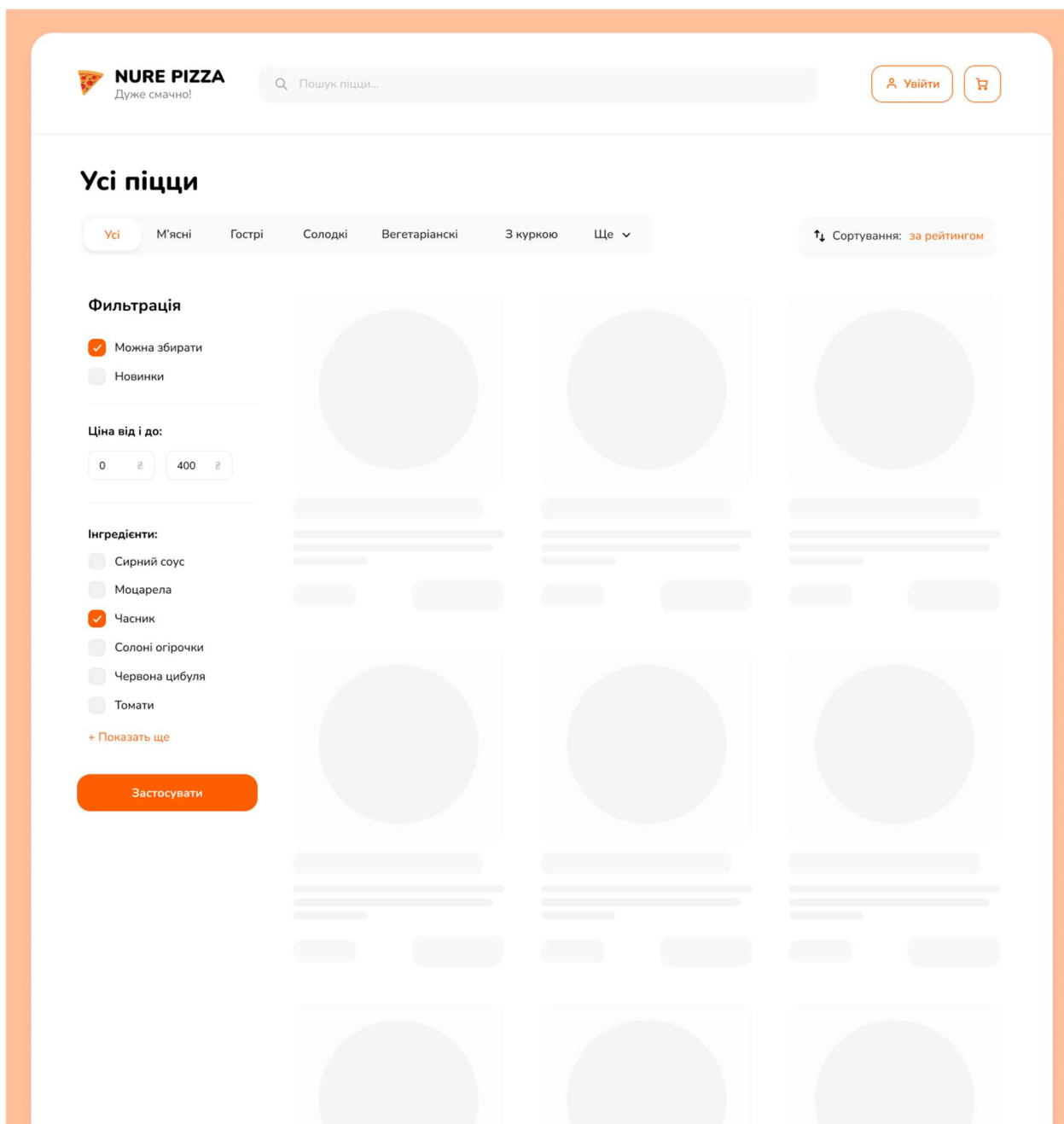


Рисунок Б.8 – Інтерфейс завантаження даних каталогу (рисунок виконано самостійно)

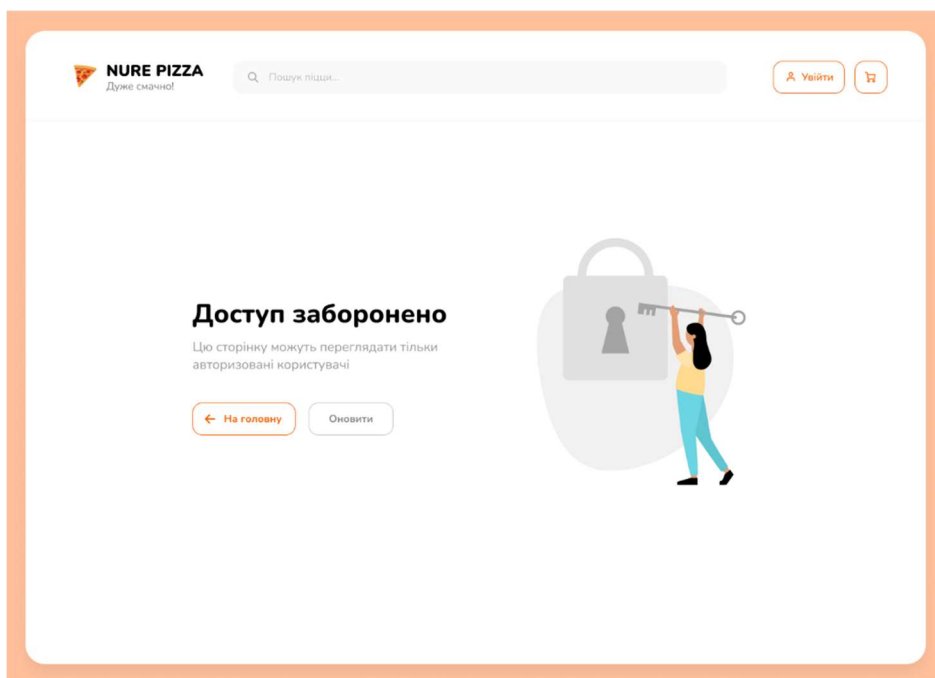


Рисунок Б.9 – Інтерфейс сторінки з повідомленням про обмеження доступу для неавторизованих користувачів (рисунок виконано самостійно)

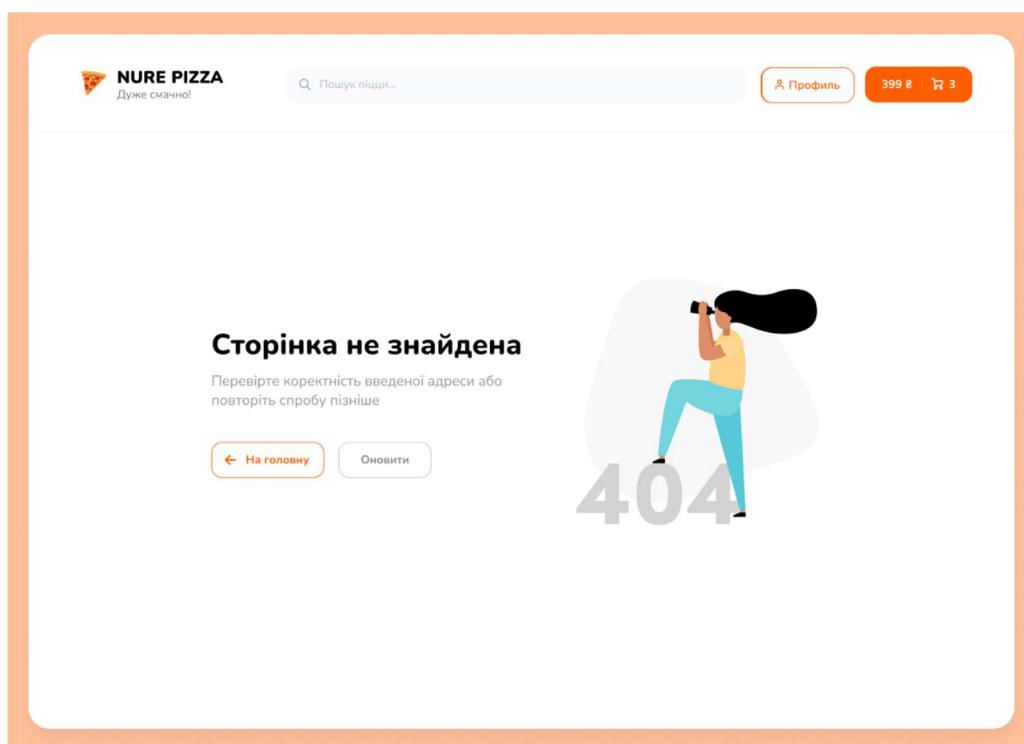


Рисунок Б.10 – Інтерфейс сторінки помилки 404 з повідомленням про відсутню сторінку (рисунок виконано самостійно)

**NEXT PIZZA**  
Дуже смачно!

[Увійти](#)

## Оформлення замовлення

### 1. Кошик

[Очистити кошик](#)

	Чизбургер-піцца Середня 30 см, традиційне тісто	838 €	- 2 + ×
	Diablo Велика 35 см, традиційне тісто + моцарела, печериці	288 €	- 1 + ×

### 2. Персональна інформація

Ім'я Вася	Прізвище <input type="text"/> <small>Введіть ваше прізвище</small>
E-Mail vasya@pupkin.ua	Телефон +380 (95) 444-22-00

### 3. Адреса доставки

Введіть адресу

- Київ, вул. Паркова, 26
- Київ, вул. Паркова, 26А
- Київ, вул. Паркова, 26/1

Время доставки  
● Час доставки ▾

**Разом:**  
**1242 €**

Стоимость товаров:	1126 €
Налог:	56 €
Доставка:	60 €

У мене є промокод

[Перейти до оплати](#)

Рисунок Б.11 – Інтерфейс форми оформлення замовлення з кошиком, персональними даними та адресою доставки (рисунок виконано самостійно)

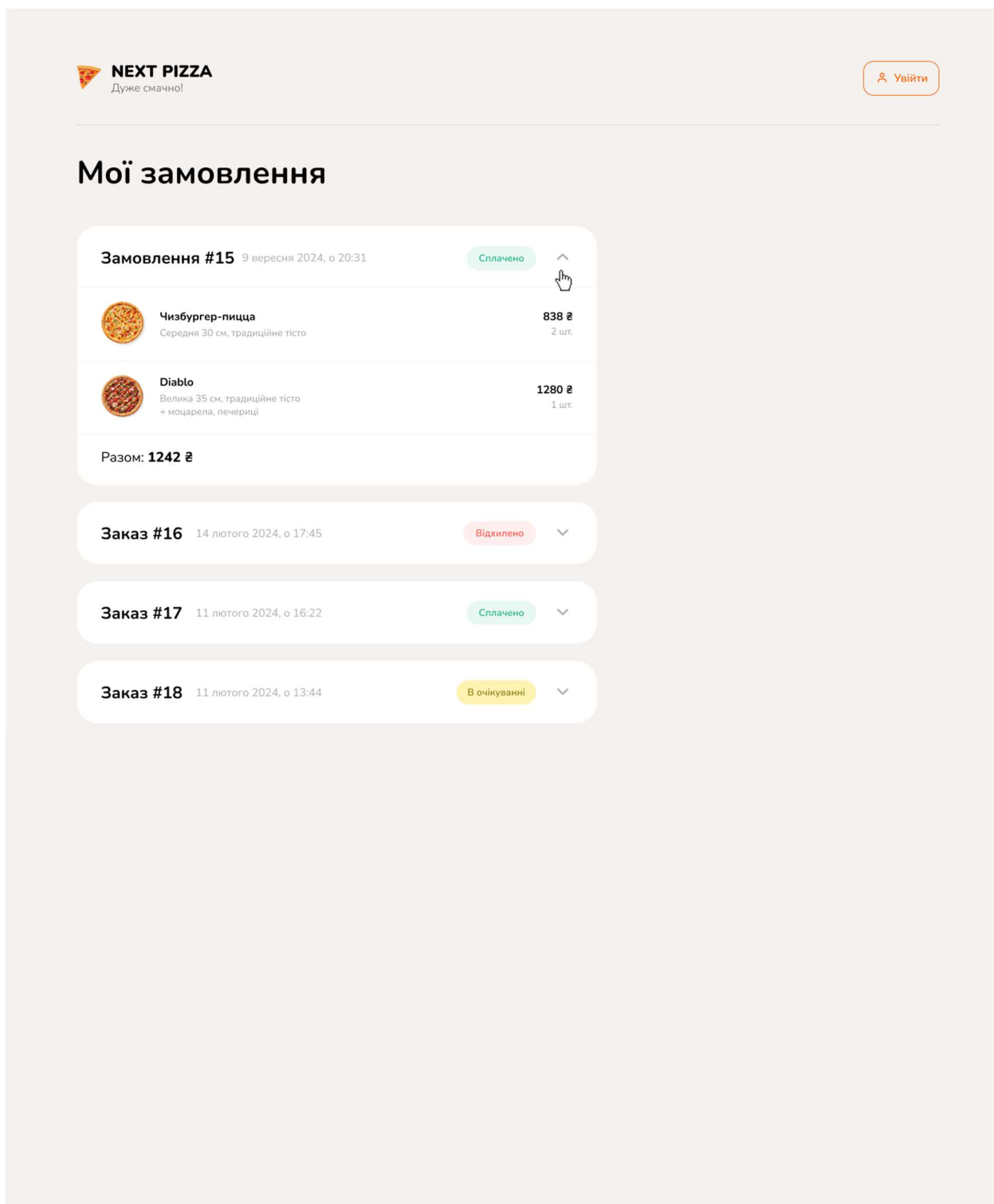


Рисунок Б.12 – Інтерфейс сторінки перегляду історії замовлень із відображенням статусу кожного замовлення (рисунок виконано самостійно)

**NEXT PIZZA**  
Дуже смачно!

Увійти

## Оформлення замовлення

### 1. Кошик

Очистити кошик

	<b>Чизбургер-піцца</b> Середня 30 см, традиційне тісто	838 ₴	- 2 +	×
	<b>Diablo</b> Велика 35 см, традиційне тісто + моцарела, печериці	288 ₴	- 1 +	×

### 2. Персональна інформація

Ім'я:

Прізвище:

Введіть ваше прізвище

E-Mail:

Телефон:

### 3. Адреса доставки

Введіть адресу:

Коментар до замовлення:

Время доставки:

Перейти до оплати

**Разом: 1242 ₴**

Стоймость товарів: 1126 ₴  
 Налоги: 56 ₴  
 Доставка: 60 ₴

У мене є промокод

Рисунок Б.13 – Інтерфейс форми оформлення замовлення з повідомленням про помилку в полі вводу (рисунок виконано самостійно)

## ДОДАТОК В

## Скрипт ініціалізації бази даних prisma/seed.ts

```

import { Prisma, PrismaClient } from '@prisma/client';
import { hashSync } from 'bcryptjs';

const prisma = new PrismaClient();

const randomDecimalNumber = (min: number, max: number) => {
  return Math.floor(Math.random() * (max - min) * 10 + min * 10) /
10;
};

const generateProductItem = ({
  productId,
  pizzaType,
  size,
}): {
  productId: number;
  pizzaType?: number;
  size?: number;
}) => {
  return {
    productId,
    price: randomDecimalNumber(190, 600),
    pizzaType,
    size,
  } as Prisma.ProductItemUncheckedCreateInput;
};

const ingredients = [
  {
    name: 'Сирний бортик',
    price: 179,
    imageUrl:
'https://cdn.dodostatic.net/static/Img/Ingredients/99f5cb91225b4875bd06a26d2e842106.png',
  },
  {
    name: 'Вершкова моцарела',
    price: 79,
    imageUrl:
'https://cdn.dodostatic.net/static/Img/Ingredients/cdea869ef287426386ed634e6099a5ba.png',
  },
  {
    name: 'Сири чедер і пармезан',
    price: 79,
    imageUrl:
'https://cdn.dodostatic.net/static/Img/Ingredients/000D3A22FA54A81411E9AFA69C1FE796',
  },
  {
    name: 'Гострий перець халапеньйо',

```

```

    price: 59,
    imageUrl:

'https://cdn.dodostatic.net/static/Img/Ingredients/11ee95b6bdfd98fb88a113db
92d7b3df.png',
  },
  {
    name: 'Ніжне курча',
    price: 79,
                                                    imageUrl:
'https://cdn.dodostatic.net/static/Img/Ingredients/000D3A39D824A82E11E9AFA5
B328D35A',
  },
  {
    name: 'Шампінйони',
    price: 59,
                                                    imageUrl:
'https://cdn.dodostatic.net/static/Img/Ingredients/000D3A22FA54A81411E9AFA6
7259A324',
  },
  {
    name: 'Шинка',
    price: 79,
                                                    imageUrl:
'https://cdn.dodostatic.net/static/Img/Ingredients/000D3A39D824A82E11E9AFA6
1B9A8D61',
  },
  {
    name: 'Пікантна пепероні',
    price: 79,
                                                    imageUrl:
'https://cdn.dodostatic.net/static/Img/Ingredients/000D3A22FA54A81411E9AFA6
258199C3',
  },
  {
    name: 'Гостра чорізо',
    price: 79,
                                                    imageUrl:
'https://cdn.dodostatic.net/static/Img/Ingredients/000D3A22FA54A81411E9AFA6
2D5D6027',
  },
  {
    name: 'Мариновані огірочки',
    price: 59,
                                                    imageUrl:
'https://cdn.dodostatic.net/static/Img/Ingredients/000D3A21DA51A81211E9EA89
958D782B',
  },
  {
    name: 'Свіжі томати',
    price: 59,
                                                    imageUrl:
'https://cdn.dodostatic.net/static/Img/Ingredients/000D3A39D824A82E11E9AFA7
AC1A1D67',
  },
  {
    name: 'Червона цибуля',

```

```

    price: 59,
    imageUrl:
'https://cdn.dodostatic.net/static/Img/Ingredients/000D3A22FA54A81411E9AFA6
0AE6464C',
  },
  {
    name: 'Соковиті ананаси',
    price: 59,
    imageUrl:
'https://cdn.dodostatic.net/static/Img/Ingredients/000D3A21DA51A81211E9AFA6
795BA2A0',
  },
  {
    name: 'Італійські трави',
    price: 39,
    imageUrl:
'https://cdn.dodostatic.net/static/Img/Ingredients/370dac9ed21e4bffa9bc261
8d258734.png',
  },
  {
    name: 'Солодкий перець',
    price: 59,
    imageUrl:
'https://cdn.dodostatic.net/static/Img/Ingredients/000D3A22FA54A81411E9AFA6
3F774C1B',
  },
  {
    name: 'Кубики бринзи',
    price: 79,
    imageUrl:
'https://cdn.dodostatic.net/static/Img/Ingredients/000D3A39D824A82E11E9AFA6
B0FFC349',
  },
  {
    name: 'Мітболи',
    price: 79,
    imageUrl:
'https://cdn.dodostatic.net/static/Img/Ingredients/b2f3a5d5afe44516a93cfc0d
2ee60088.png',
  },
].map((obj, index) => ({ id: index + 1, ...obj }));

async function up() {
  await prisma.user.createMany({
    data: [
      {
        fullName: 'User Test',
        email: 'user@test.ua',
        password: hashSync('111111', 10),
        verified: new Date(),
        role: 'USER',
      },
      {
        fullName: 'Admin Test',

```

```

        email: 'admin@test.ua',
        password: hashSync('111111', 10),
        verified: new Date(),
        role: 'ADMIN',
    },
],
});

await prisma.category.createMany({
  data: [
    {
      name: 'Піцци',
    },
    {
      name: 'Сніданок',
    },
    {
      name: 'Закуски',
    },
    {
      name: 'Коктейлі',
    },
    {
      name: 'Напої',
    },
  ],
});

await prisma.product.createMany({
  data: [
    {
      name: 'Омлет з шинкою та грибами',
      imageUrl:
'https://media.dodostatic.net/image/r:292x292/11EE7970321044479C1D1085457A36EB.webp',
      categoryId: 2,
    },
    {
      name: 'Омлет з пепероні',
      imageUrl:
'https://media.dodostatic.net/image/r:292x292/11EE94ECF33B0C46BA410DEC1B1DD6F8.webp',
      categoryId: 2,
    },
    {
      name: 'Кава Латте',
      imageUrl:
'https://media.dodostatic.net/image/r:292x292/11EE7D61B0C26A3F85D97A78FEEE00AD.webp',
      categoryId: 2,
    },
    {
      name: 'Сендвіч шинка та сир',
      imageUrl:

```

```
'https://media.dodostatic.net/image/r:292x292/11EE796FF0059B799A17F57A9E64C725.webp',
  categoryId: 3,
},
{
  name: 'Курячі нагетси',
  imageUrl:

'https://media.dodostatic.net/image/r:292x292/11EE7D618B5C7EC29350069AE9532C6E.webp',
  categoryId: 3,
},
{
  name: 'Картопля з печі з соусом 🍷',
  imageUrl:

'https://media.dodostatic.net/image/r:292x292/11EED646A9CD324C962C6BEA78124F19.webp',
  categoryId: 3,
},
{
  name: 'Додстер',
  imageUrl:

'https://media.dodostatic.net/image/r:292x292/11EE796F96D11392A2F6DD73599921B9.webp',
  categoryId: 3,
},
{
  name: 'Гострий Додстер 🌶️🌶️',
  imageUrl:

'https://media.dodostatic.net/image/r:292x292/11EE796FD3B594068F7A752DF8161D04.webp',
  categoryId: 3,
},
{
  name: 'Банановий молочний коктейль',
  imageUrl:

'https://media.dodostatic.net/image/r:292x292/11EEE20B8772A72A9B60CFB20012C185.webp',
  categoryId: 4,
},
{
  name: 'Карамельне яблуко молочний коктейль',
  imageUrl:

'https://media.dodostatic.net/image/r:292x292/11EE79702E2A22E693D96133906FB1B8.webp',
  categoryId: 4,
},
{
  name: 'Молочний коктейль з печивом Орео',
  imageUrl:
```

```

'https://media.dodostatic.net/image/r:292x292/11EE796FA1F50F8F8111A399E4C1A
1E3.webp',
    categoryId: 4,
  },
  {
    name: 'Класичний молочний коктейль 🍹',
    imageUrl:

'https://media.dodostatic.net/image/r:292x292/11EE796F93FB126693F96CB1D3E40
3FB.webp',
    categoryId: 4,
  },
  {
    name: 'Ірландський капучино',
    imageUrl:

'https://media.dodostatic.net/image/r:292x292/11EE7D61999EBDA59C10E216430A6
093.webp',
    categoryId: 5,
  },
  {
    name: 'Кава карамельний капучино',
    imageUrl:

'https://media.dodostatic.net/image/r:292x292/11EE7D61AED6B6D4BFDAD4E58D76C
F56.webp',
    categoryId: 5,
  },
  {
    name: 'Кава кокосовий латте',
    imageUrl:

'https://media.dodostatic.net/image/r:292x292/11EE7D61B19FA07090EE88B0ED347
F42.webp',
    categoryId: 5,
  },
  {
    name: 'Кава Американо',
    imageUrl:

'https://media.dodostatic.net/image/r:292x292/11EE7D61B044583596548A59078BB
D33.webp',
    categoryId: 5,
  },
  {
    name: 'Кава Латте',
    imageUrl:

'https://media.dodostatic.net/image/r:292x292/11EE7D61B0C26A3F85D97A78FEEE0
0AD.webp',
    categoryId: 5,
  },
],
});

await prisma.ingredient.createMany({

```

```

    data: ingredients,
  });

  const pizza1 = await prisma.product.create({
    data: {
      name: 'Пеperони фреш',
      imageUrl:

'https://media.dodostatic.net/image/r:233x233/11EE7D61304FAF5A98A6958F2BB2D
260.webp',
      categoryId: 1,
      ingredients: {
        connect: ingredients.slice(0, 5),
      },
    },
  });

  const pizza2 = await prisma.product.create({
    data: {
      name: 'Сирна',
      imageUrl:

'https://media.dodostatic.net/image/r:233x233/11EE7D610CF7E265B7C72BE5AE757
CA7.webp',
      categoryId: 1,
      ingredients: {
        connect: ingredients.slice(5, 10),
      },
    },
  });

  const pizza3 = await prisma.product.create({
    data: {
      name: 'Чоризо фреш',
      imageUrl:

'https://media.dodostatic.net/image/r:584x584/11EE7D61706D472F9A5D71EB94149
304.webp',
      categoryId: 1,
      ingredients: {
        connect: ingredients.slice(10, 40),
      },
    },
  });

  await prisma.productItem.createMany({
    data: [
      // Піца "Пеperони фреш"
      generateProductItem({ productId: pizza1.id, pizzaType: 1, size:
20 }),
      generateProductItem({ productId: pizza1.id, pizzaType: 2, size:
30 }),
      generateProductItem({ productId: pizza1.id, pizzaType: 2, size:
40 }),

      // Піца "Сирна"

```

```

    generateProductItem({ productId: pizza2.id, pizzaType: 1, size:
20  } ),
    generateProductItem({ productId: pizza2.id, pizzaType: 1, size:
30  } ),
    generateProductItem({ productId: pizza2.id, pizzaType: 1, size:
40  } ),
    generateProductItem({ productId: pizza2.id, pizzaType: 2, size:
20  } ),
    generateProductItem({ productId: pizza2.id, pizzaType: 2, size:
30  } ),
    generateProductItem({ productId: pizza2.id, pizzaType: 2, size:
40  } ),

    // Піца "Чорізо фреш"
    generateProductItem({ productId: pizza3.id, pizzaType: 1, size:
20  } ),
    generateProductItem({ productId: pizza3.id, pizzaType: 2, size:
30  } ),
    generateProductItem({ productId: pizza3.id, pizzaType: 2, size:
40  } ),

    // Інші продукти
    generateProductItem({ productId: 1  } ),
    generateProductItem({ productId: 2  } ),
    generateProductItem({ productId: 3  } ),
    generateProductItem({ productId: 4  } ),
    generateProductItem({ productId: 5  } ),
    generateProductItem({ productId: 6  } ),
    generateProductItem({ productId: 7  } ),
    generateProductItem({ productId: 8  } ),
    generateProductItem({ productId: 9  } ),
    generateProductItem({ productId: 10 } ),
    generateProductItem({ productId: 11 } ),
    generateProductItem({ productId: 12 } ),
    generateProductItem({ productId: 13 } ),
    generateProductItem({ productId: 14 } ),
    generateProductItem({ productId: 15 } ),
    generateProductItem({ productId: 16 } ),
    generateProductItem({ productId: 17 } ),
  ],
});

await prisma.cart.createMany({
  data: [
    {
      userId: 1,
      totalAmount: 0,
    },
    {
      userId: 2,
      totalAmount: 0,
    },
  ],
});

await prisma.cartItem.create({
  data: {

```

```

        productId: 1,
        cartId: 1,
        userId: 1,
        quantity: 1,
        pizzaSize: 20,
        ingredients: {
          connect: [{ id: 1 }, { id: 2 }, { id: 3 }],
        },
      },
    });

await prisma.story.createMany({
  data: [
    {
      previewImageUrl:
        'https://img.freepik.com/premium-vector/food-menu-delicious-pizza-instagram-story-template_565017-814.jpg?w=2000',
    },
    {
      previewImageUrl:
        'https://img.freepik.com/premium-psd/delicious-pizza-instagram-story-template-design_440834-443.jpg?w=826',
    },
    {
      previewImageUrl:
        'https://img.freepik.com/premium-psd/delicious-pizza-instagram-story-template-design_440834-443.jpg?w=826',
    },
    {
      previewImageUrl:
        'https://img.freepik.com/premium-vector/food-menu-delicious-pizza-instagram-story-template_565017-814.jpg?w=2000',
    },
    {
      previewImageUrl:
        'https://img.freepik.com/premium-psd/pizza-instagram-stories-banner-template_65983-1333.jpg?w=1380',
    },
    {
      previewImageUrl:
        'https://img.freepik.com/premium-psd/delicious-pizza-instagram-story-template-design_440834-443.jpg?w=826',
    },
  ],
});

await prisma.storyItem.createMany({
  data: [
    {
      storyId: 1,
      sourceUrl:
        'https://img.freepik.com/premium-vector/food-menu-delicious-pizza-instagram-story-template_565017-814.jpg?w=2000',
    },
    {
      storyId: 1,
      sourceUrl:

```

```

        'https://thumbs.dreamstime.com/b/savor-stretch-cheese-four-
cheese-italian-pizza-savor-stretch-cheese-four-cheese-italian-pizza-
vertical-294185801.jpg?w=576',
    },
    {
        storyId: 1,
        sourceUrl:
            'https://img.freepik.com/premium-psd/delicious-pizza-
instagram-story-template-design_440834-443.jpg?w=826',
    },
    {
        storyId: 1,
        sourceUrl:
            'https://img.freepik.com/premium-vector/food-menu-
delicious-pizza-instagram-story-template_565017-814.jpg?w=2000',
    },
    {
        storyId: 1,
        sourceUrl:
            'https://thumbs.dreamstime.com/b/savor-stretch-cheese-four-
cheese-italian-pizza-savor-stretch-cheese-four-cheese-italian-pizza-
vertical-294185801.jpg?w=576',
    },
    ],
    });
}

```

```

}

async function down() {

    const tables = [
        'User',
        'Product',
        'Ingredient',
        'Order',
        'ProductItem',
        'VerificationCode',
        'Category',
        'Cart',
        'CartItem',
        'Story',
        'StoryItem',
    ];

    await prisma.$executeRaw`TRUNCATE TABLE "Product" RESTART IDENTITY
CASCADE;`;
    await prisma.$executeRaw`TRUNCATE TABLE "Ingredient" RESTART
IDENTITY CASCADE;`;
    await prisma.$executeRaw`TRUNCATE TABLE "Order" RESTART IDENTITY
CASCADE;`;
    await prisma.$executeRaw`TRUNCATE TABLE "ProductItem" RESTART
IDENTITY CASCADE;`;
    await prisma.$executeRaw`TRUNCATE TABLE "VerificationCode" RESTART
IDENTITY CASCADE;`;
    await prisma.$executeRaw`TRUNCATE TABLE "Category" RESTART IDENTITY
CASCADE;`;
}

```

```
    await prisma.$executeRaw`TRUNCATE TABLE "Cart" RESTART IDENTITY
CASCADE;`;
    await prisma.$executeRaw`TRUNCATE TABLE "CartItem" RESTART IDENTITY
CASCADE;`;
    await prisma.$executeRaw`TRUNCATE TABLE "Story" RESTART IDENTITY
CASCADE;`;
    await prisma.$executeRaw`TRUNCATE TABLE "StoryItem" RESTART
IDENTITY CASCADE;`;
    await prisma.$executeRaw`TRUNCATE TABLE "User" RESTART IDENTITY
CASCADE;`;
  }

  async function main() {
    try {
      // await down();
      await up();
    } catch (e) {
      console.error(e);
    }
  }

  main()
    .then(async () => {
      await prisma.$disconnect();
    })
    .catch(async (e) => {
      console.error(e);
      await prisma.$disconnect();
      process.exit(1);
    });
}
```

## ДОДАТОК Г

## Програмний код для інтеграції з LIQPAY

```

import crypto from 'crypto';

interface Props {
  description: string;
  orderId: number;
  amount: number;
}

export async function createLiqPayPayment(details: Props) {
  const public_key = process.env.LIQPAY_PUBLIC_KEY!;
  const private_key = process.env.LIQPAY_PRIVATE_KEY!;

  const params = {
    version: '3',
    public_key,
    action: 'pay',
    amount: details.amount.toString(),
    currency: 'UAH',
    description: details.description,
    order_id: details.orderId.toString(),
    language: 'ua',
    result_url: process.env.LIQPAY_RESULT_URL,
    server_url: process.env.LIQPAY_CALLBACK_URL,
  };

  const data = Buffer.from(JSON.stringify(params)).toString('base64');

  // Підпис: SHA1(private_key + data + private_key), потім base64
  const signature = crypto
    .createHash('sha1')
    .update(private_key + data + private_key)
    .digest('base64');

  const paymentUrl = `https://www.liqpay.ua/api/3/checkout?data=${data}&signature=${signature}`;

  return { data, signature, paymentUrl };
}

```

## ДОДАТОК Д

Програмний код для інтеграції з google-maps та google-autocomplete для вибору адреси

```

'use client';

import React, { useRef, useState } from 'react';
import Autocomplete from 'react-google-autocomplete';
import { GoogleMap, Marker, useJsApiLoader } from '@react-google-
maps/api';

interface Props {
  onChange?: (value?: string) => void;
}

const containerStyle = {
  width: '100%',
  height: '300px',
};

export const AdressInput: React.FC<Props> = ({ onChange }) => {
  const [location, setLocation] = useState<{ lat: number; lng:
number } | null>(null);
  const [address, setAddress] = useState<string>('');
  const autocompleteRef = useRef<HTMLInputElement>(null);

  const { isLoading } = useJsApiLoader({
    googleMapsApiKey: process.env.NEXT_PUBLIC_GOOGLE_API_KEY!,
    libraries: ['places'],
    language: 'uk',
  });

  const handlePlaceSelect = (place: google.maps.places.PlaceResult)
=> {
    const formattedAddress = place.formatted_address;
    const lat = place.geometry?.location?.lat();
    const lng = place.geometry?.location?.lng();

    if (lat && lng) {
      setLocation({ lat, lng });
    }

    if (formattedAddress) {
      setAddress(formattedAddress);
      onChange?.(formattedAddress);
    }
  };

  const handleMapClick = async (e: google.maps.MapMouseEvent) => {
    if (e.latLng) {
      const lat = e.latLng.lat();
      const lng = e.latLng.lng();
      setLocation({ lat, lng });

      const geocoder = new google.maps.Geocoder();

```

```

status) => {
    geocoder.geocode({ location: { lat, lng } }, (results,
    if (status === 'OK' && results?.[0]) {
        const resultAddress = results[0].formatted_address;
        setAddress(resultAddress);
        onChange?.(resultAddress);
        if (autocompleteRef.current) {
            autocompleteRef.current.value = resultAddress;
        }
    }
    });
};

if (!isLoading) return <p>Завантаження карти...</p>;

return (
    <div className="flex flex-col gap-4">
        <Autocomplete
            apiKey={process.env.NEXT_PUBLIC_GOOGLE_API_KEY}
            onPlaceSelected={handlePlaceSelect}
            options={{
                types: ['address'],
                componentRestrictions: { country: 'ua' },
            }}
            className="text-base w-full border px-4 py-2 rounded-xl"
            placeholder="Введіть адресу..."
            ref={autocompleteRef}
        />

        {location && (
            <GoogleMap
                mapContainerStyle={containerStyle}
                center={location}
                zoom={15}
                onClick={handleMapClick}
            >
                <Marker position={location} />
            </GoogleMap>
        )}
    </div>
);
};

```

## ДОДАТОК Е

## Програмний код для реєстрації та авторизації користувача за допомогою Google OAuth 2.0

```

import { AuthOptions } from 'next-auth';
import CredentialsProvider from 'next-auth/providers/credentials';
import GoogleProvider from 'next-auth/providers/google';

import { prisma } from '@/prisma/prisma-client';
import { compare, hashSync } from 'bcryptjs';

export const authOptions: AuthOptions = {
  providers: [
    GoogleProvider({
      clientId: process.env.GOOGLE_CLIENT_ID || '',
      clientSecret: process.env.GOOGLE_CLIENT_SECRET || '',
    }),
    CredentialsProvider({
      name: 'Credentials',
      credentials: {
        email: { label: 'Email', type: 'text' },
        password: { label: 'Password', type: 'password' },
      },
      async authorize(credentials) {
        if (!credentials) return null;

        const findUser = await prisma.user.findFirst({
          where: { email: credentials.email },
        });

        if (!findUser) return null;

        const isPasswordValid = await compare(credentials.password,
findUser.password);
        if (!isPasswordValid || !findUser.verified) return null;

        return {
          id: findUser.id,
          email: findUser.email,
          name: findUser.fullName,
          role: findUser.role,
        };
      },
    }),
  ],
  secret: process.env.NEXTAUTH_SECRET,
  session: {
    strategy: 'jwt',
  },
  callbacks: {
    async signIn({ user, account }) {
      try {
        if (account?.provider === 'credentials') return true;

        if (!user.email) return false;

```

```

const findUser = await prisma.user.findFirst({
  where: {
    OR: [
      { provider: account?.provider, providerId:
account?.providerAccountId },
      { email: user.email },
    ],
  },
});

if (findUser) {
  await prisma.user.update({
    where: { id: findUser.id },
    data: {
      provider: account?.provider,
      providerId: account?.providerAccountId,
    },
  });
  return true;
}

await prisma.user.create({
  data: {
    email: user.email,
    fullName: user.name || 'User #' + user.id,
    password: hashSync(user.id.toString(), 10),
    verified: new Date(),
    provider: account?.provider,
    providerId: account?.providerAccountId,
  },
});

return true;
} catch (error) {
  console.error('Error [SIGNIN]', error);
  return false;
}
},
async jwt({ token }) {
  if (!token.email) return token;

  const findUser = await prisma.user.findFirst({
    where: { email: token.email },
  });

  if (findUser) {
    token.id = String(findUser.id);
    token.email = findUser.email;
    token.fullName = findUser.fullName;
    token.role = findUser.role;
  }

  return token;
},
session({ session, token }) {
  if (session?.user) {

```

```
        session.user.id = token.id;
        session.user.role = token.role;
    }

    return session;
},
},
};
```

## ДОДАТОК Ж

Програмний код логіки для оновлення інформації користувача, його авторизації та створення кошику замовлення

```

'use server';

import { prisma } from '@prisma/prisma-client';
import { PayOrderTemplate } from '@shared/components';
import { VerificationUserTemplate } from '@shared/components/shared/email-templates/verification-user';
import { CheckoutFormValues } from '@shared/constants';
import { createLiqPayPayment, sendEmail } from '@shared/lib';
import { getUserSession } from '@shared/lib/get-user-session';
import { OrderStatus, Prisma } from '@prisma/client';
import { hashSync } from 'bcryptjs';
import { cookies } from 'next/headers';

export async function createOrder(data: CheckoutFormValues) {
  try {
    const sessionUser = await getUserSession();
    if (!sessionUser) {
      throw new Error('Користувач не авторизований');
    }

    const userId = Number(sessionUser.id);

    const cookieStore = await cookies();
    const cartToken = cookieStore.get('cartToken')?.value;

    if (!cartToken) {
      throw new Error('Токен кошика не знайдено');
    }

    const userCart = await prisma.cart.findFirst({
      include: {
        user: true,
        items: {
          include: {
            ingredients: true,
            productItem: {
              include: {
                product: true,
              },
            },
          },
        },
      },
      where: {
        OR: [
          { userId },
          { tokenId: cartToken },
        ],
      },
    });
  }
};

```

```

if (!userCart) {
  throw new Error('Кошик не знайдено');
}

if (userCart.totalAmount === 0) {
  throw new Error('Кошик порожній');
}

const order = await prisma.order.create({
  data: {
    userId,
    tokenId: cartToken,
    fullName: `${data.firstName} ${data.lastName}`,
    email: data.email,
    phone: data.phone,
    address: data.address,
    comment: data.comment,
    totalAmount: userCart.totalAmount,
    status: OrderStatus.PENDING,
    items: JSON.stringify(userCart.items),
  },
});

await prisma.cart.update({
  where: { id: userCart.id },
  data: { totalAmount: 0 },
});

await prisma.cartItem.deleteMany({
  where: { cartId: userCart.id },
});

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

if (!paymentData) throw new Error('Не вдалося створити платіж');

await prisma.order.update({
  where: { id: order.id },
  data: { paymentId: paymentData.signature },
});

await sendEmail(
  data.email,
  `Next Pizza / Оплатіть замовлення №${order.id}`,
  PayOrderTemplate({
    orderId: order.id,
    totalAmount: order.totalAmount,
    paymentUrl: paymentData.paymentUrl,
  })
);

return paymentData.paymentUrl;
} catch (err) {

```

```

        console.error('[CreateOrder] Server error', err);
        throw err;
    }
}

export async function updateUserInfo(body: Prisma.UserUpdateInput) {
    try {
        const currentUser = await getUserSession();

        if (!currentUser) {
            throw new Error('Користувача не знайдено');
        }

        const currentUserId = Number(currentUser.id);

        const existing = await prisma.user.findUnique({
            where: { id: currentUserId },
        });

        if (!existing) {
            throw new Error('Користувача не знайдено');
        }

        await prisma.user.update({
            where: { id: currentUserId },
            data: {
                fullName: body.fullName,
                email: body.email,
                password: body.password
                    ? hashSync(body.password as string, 10)
                    : existing.password,
            },
        });
    } catch (err) {
        console.log('[ОновленняКористувача] Помилка', err);
        throw err;
    }
}

export async function registerUser(body: Prisma.UserCreateInput) {
    try {
        const user = await prisma.user.findUnique({
            where: { email: body.email },
        });

        if (user) {
            if (!user.verified) {
                throw new Error('Електронну адресу не підтверджено');
            }
            throw new Error('Користувач вже існує');
        }

        const createdUser = await prisma.user.create({
            data: {
                fullName: body.fullName,
                email: body.email,
                password: hashSync(body.password, 10),
            },
        });
    }
}

```

```
        verified: new Date(),
    },
    });

    const code = Math.floor(100000 + Math.random() *
900000).toString();

    await prisma.verificationCode.create({
      data: {
        code,
        userId: createdUser.id,
      },
    });

    await sendEmail(
      createdUser.email,
      'Next Pizza / Підтвердження реєстрації',
      VerificationUserTemplate({ code })
    );
  } catch (err) {
    console.log('[РеєстраціяКористувача] Помилка', err);
    throw err;
  }
}
```

## ДОДАТОК К

Маршрути для оновлення та видалення позицій у кошику

```

import { prisma } from '@shared/lib/prisma';
import { updateCartTotalAmount } from '@shared/lib';
import { NextRequest, NextResponse } from 'next/server';

export async function PATCH(req: NextRequest, { params }: { params: {
id: string } }) {
  try {
    const id = Number(params.id);
    const data = (await req.json()) as { quantity: number };
    const tokenId = req.cookies.get('cartToken')?.value;

    if (!tokenId) {
      return NextResponse.json({ error: 'Cart token not found' });
    }

    const cartItem = await prisma.cartItem.findFirst({
      where: {
        id,
      },
    });

    if (!cartItem) {
      return NextResponse.json({ error: 'Cart item not found' });
    }

    await prisma.cartItem.update({
      where: {
        id,
      },
      data: {
        quantity: data.quantity,
      },
    });

    const updatedUserCart = await updateCartTotalAmount(tokenId);
    return NextResponse.json(updatedUserCart);
  } catch (error) {
    console.log('[CART_PATCH] Server error', error);
    return NextResponse.json(
      { message: 'Не вдалося оновити кошик',
        status: 500 }
    );
  }
}

export async function DELETE(req: NextRequest, { params }: { params:
{ id: string } }) {
  try {
    const id = Number(params.id);
    const tokenId = req.cookies.get('cartToken')?.value;

    if (!tokenId) {
      return NextResponse.json({ error: 'Cart token not found' });
    }

```

```
    }

    const cartItem = await prisma.cartItem.findFirst({
      where: {
        id: id,
      },
    });

    if (!cartItem) {
      return NextResponse.json({ error: 'Cart item not found' });
    }

    await prisma.cartItem.delete({
      where: {
        id: id,
      },
    });
    const updatedUserCart = await updateCartTotalAmount(tokenId);
    return NextResponse.json(updatedUserCart);

  } catch (error) {
    console.log('[CART_DELETE] Server error', error);
    return NextResponse.json({ message: 'Не удалось видалити корзину'
}, { status: 500 }
    );
  }
}
```

## ДОДАТОК Л

Серверні дії: реєстрація з email-токеном, оновлення профілю, створення замовлення, інтеграція LiqPay та email-квитанції.

```
'use server';

import { prisma } from '@prisma/prisma-client';
import { PayOrderTemplate } from '@shared/components';
import { VerificationUserTemplate } from '@shared/components/shared/email-templates/verification-user';
import { CheckoutFormValues } from '@shared/constants';
import { createLiqPayPayment, sendEmail } from '@shared/lib';
import { getUserSession } from '@shared/lib/get-user-session';
import { OrderStatus, Prisma } from '@prisma/client';
import { hashSync } from 'bcryptjs';
import { cookies } from 'next/headers';

export async function createOrder(data: CheckoutFormValues) {
  try {
    const sessionUser = await getUserSession();
    if (!sessionUser) {
      throw new Error('Користувач не авторизований');
    }

    const userId = Number(sessionUser.id);

    const cookieStore = await cookies();
    const cartToken = cookieStore.get('cartToken')?.value;

    if (!cartToken) {
      throw new Error('Токен кошика не знайдено');
    }

    const userCart = await prisma.cart.findFirst({
      include: {
        user: true,
        items: {
          include: {
            ingredients: true,
            productItem: {
              include: {
                product: true,
              },
            },
          },
        },
      },
      where: {
        OR: [
          { userId },
          { tokenId: cartToken },
        ],
      },
    });
  }
}
```

```

if (!userCart) {
  throw new Error('Кошик не знайдено');
}

if (userCart.totalAmount === 0) {
  throw new Error('Кошик порожній');
}

const order = await prisma.order.create({
  data: {
    userId,
    tokenId: cartToken,
    fullName: `${data.firstName} ${data.lastName}`,
    email: data.email,
    phone: data.phone,
    address: data.address,
    comment: data.comment,
    totalAmount: userCart.totalAmount,
    status: OrderStatus.PENDING,
    items: JSON.stringify(userCart.items),
  },
});

await prisma.cart.update({
  where: { id: userCart.id },
  data: { totalAmount: 0 },
});

await prisma.cartItem.deleteMany({
  where: { cartId: userCart.id },
});

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

if (!paymentData) throw new Error('Не вдалося створити платіж');

await prisma.order.update({
  where: { id: order.id },
  data: { paymentId: paymentData.signature },
});

await sendEmail(
  data.email,
  `Next Pizza / Оплатіть замовлення №${order.id}`,
  PayOrderTemplate({
    orderId: order.id,
    totalAmount: order.totalAmount,
    paymentUrl: paymentData.paymentUrl,
  })
);

return paymentData.paymentUrl;
} catch (err) {

```

```

        console.error('[CreateOrder] Server error', err);
        throw err;
    }
}

export async function updateUserInfo(body: Prisma.UserUpdateInput) {
    try {
        const currentUser = await getUserSession();

        if (!currentUser) {
            throw new Error('Користувача не знайдено');
        }

        const currentUserId = Number(currentUser.id);

        const existing = await prisma.user.findUnique({
            where: { id: currentUserId },
        });

        if (!existing) {
            throw new Error('Користувача не знайдено');
        }

        await prisma.user.update({
            where: { id: currentUserId },
            data: {
                fullName: body.fullName,
                email: body.email,
                password: body.password
                    ? hashSync(body.password as string, 10)
                    : existing.password,
            },
        });
    } catch (err) {
        console.log('[ОновленняКористувача] Помилка', err);
        throw err;
    }
}

export async function registerUser(body: Prisma.UserCreateInput) {
    try {
        const user = await prisma.user.findUnique({
            where: { email: body.email },
        });

        if (user) {
            if (!user.verified) {
                throw new Error('Електронну адресу не підтверджено');
            }
            throw new Error('Користувач вже існує');
        }

        const createdUser = await prisma.user.create({
            data: {
                fullName: body.fullName,
                email: body.email,
                password: hashSync(body.password, 10),
            },
        });
    }
}

```

```
        verified: new Date(),
    },
  });

    const code = Math.floor(100000 + Math.random() *
900000).toString();

    await prisma.verificationCode.create({
      data: {
        code,
        userId: createdUser.id,
      },
    });

    await sendEmail(
      createdUser.email,
      'Next Pizza / Підтвердження реєстрації',
      VerificationUserTemplate({ code })
    );
  } catch (err) {
    console.log('[РеєстраціяКористувача] Помилка', err);
    throw err;
  }
}
```

## ДОДАТОК М

## Тестування навантаження для серверу

```
const { PrismaClient } = require('@prisma/client');
const prisma = new PrismaClient();

async function testPerformance() {
  const startTime = Date.now();

  const productsWithMozzarella = await prisma.product.findMany({
    where: {
      ingredients: {
        some: { name: "Сир Моцарела" }
      }
    },
    include: {
      // Фільтруємо лише інгредієнти з ім'ям "Сир Моцарела"
      ingredients: {
        where: { name: "Сир Моцарела" }
      }
    }
  });

  const duration = Date.now() - startTime;
  console.log("Час виконання Prisma-запиту:", duration, "ms");
  console.log(JSON.stringify(productsWithMozzarella, null, 2));
}

testPerformance()
  .catch((e) => {
    console.error(e);
  })
  .finally(async () => {
    await prisma.$disconnect();
  });
}
```

## ДОДАТОК Н

### Слайди презентації



Сервіс для вирішення  
питань пов'язаних із  
замовленнями в піцерії



Мартинов Богдан Валерійович, ПЗПІ-217  
Керівник: доцент Ірина ГУЗДО

17червня 2025

Рисунок Н.1 – Слайд

## Мета роботи

Мета кваліфікаційної роботи полягає у розробці сучасного веб-сервісу для замовлень у піцерії, який забезпечить зручний та швидкий спосіб оформлення замовлень і оптимізує процес взаємодії між закладом і клієнтами.

Актуальність роботи:

- Стрімке зростання ринку онлайн-замовлень їжі.
- Користувачі очікують кастомізації та миттєвих сповіщень.
- Бракує локальних рішень із підтримкою UA-платежів.
- Автоматизація скорочує витрати бізнесу та час клієнта.



Рисунок Н.2 – Слайд 2

## Аналіз проблеми (аналіз існуючих рішень)

### • Досліджені сервіси

- Domino's Pizza
- Papa John's
- Pizza 33
- Mister Cat
- Il Molino

### Виявлені прогалини

- ✗ Нема онлайн-оплати
- ✗ Обмежена функціональність веб-застосунку
- ✗ Застарілий інтерфейс
- ✗ Соціальні мережи

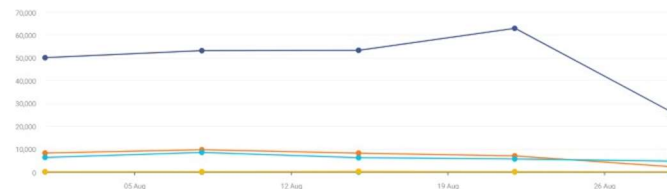


3

Рисунок Н.3 – Слайд 3

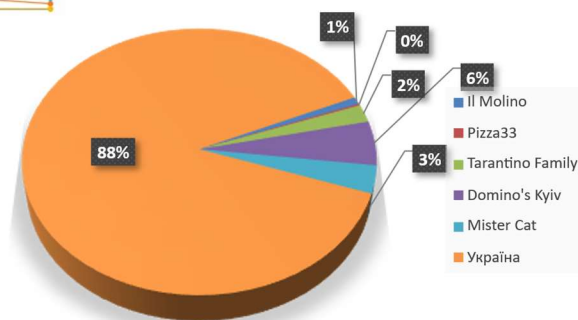
## Аналіз проблеми (аналіз існуючих рішень)

dominos.ua 245,842 tarantino-family.com 35,839 pizza33.com.ua 294 mistercat.com.ua 983 ilmolino.ua 31,938



Кількість користувачів по Україні за місяць

Продажі, млн.



Продажі ресторанів громадського харчування по всій Україні



4

Рисунок Н.4 – Слайд 4

## Постановка задачі та опис системи

### Чітке формулювання проблеми

Онлайн-замовлення у невеликих піцеріях досі здійснюють телефоном або через застарілі сайти

Немає швидкої онлайн-оплати та автоматичних сповіщень

Користувач не може сворити власну піцу під себе

Персонал витрачає час на ручну обробку та уточнення замовлень

### Опис очікуваних результатів

- ✓ приймає й оплачує замовлення без дзвінків
- ✓ дозволяє конструювати піцу й одразу бачити ціну
- ✓ інтеграція з стороніми сервісами



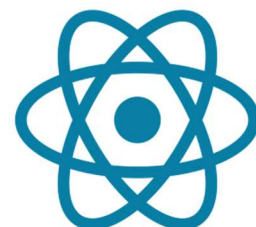
5

Рисунок Н.5 – Слайд 5

## Вибір технологій розробки



Шар	Технологія	Чому обрав
Фронт-енд	Next.js + Tailwind CSS	SSR → швидкий старт сторінки, Tailwind – легко підтримувати стиль
Бек-енд / API	Next.js API Routes	Один репо → спрощений DevOps, автоскейл на Vercel
База даних	PostgreSQL + Prisma ORM	SQL-потужність, типи в TypeScript, контроль міграцій
Аутентифікація	NextAuth.js: • API Google OAuth 2.0 • Email + Password	Користувач може зайти одним кліком через Google; для локальних акаунтів – власна форма з підтвердженням пошти
Стан клієнта	Zustand	Маленький, без «Redux-шуму»
Оплата	LiqPay API	Локальна система, низькі комісії
Адреси	Google Maps + Places	Автодоповнення, координати в замовленні



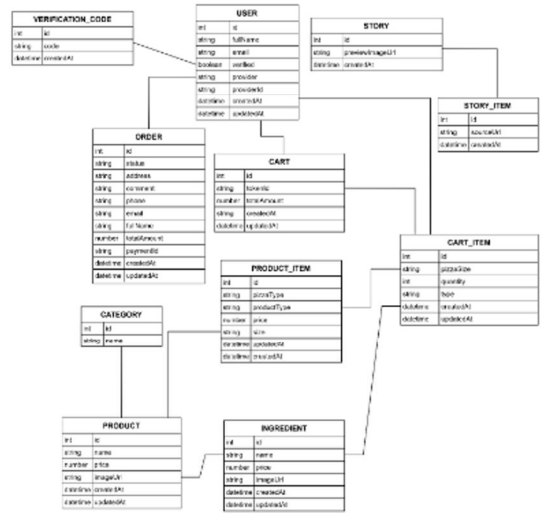
6

Рисунок Н.6 – Слайд 6

# Архітектура створення програмного забезпечення



Діаграма класів



ER-діаграма для сутностей



Рисунок Н.7 – Слайд 7

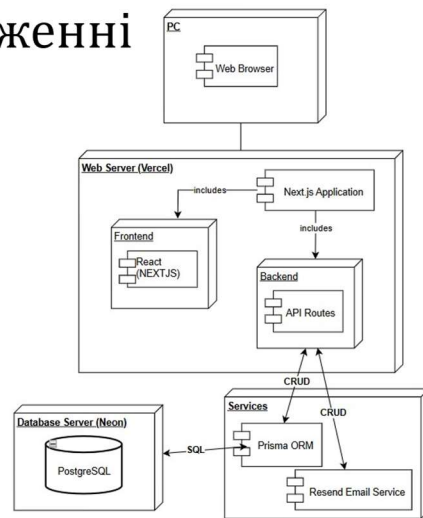
## Опис програмного забезпечення, що було використано у дослідженні

### Мови

- TypeScript** — фронт + бек
- JavaScript** — екосистема прм
- SQL (PostgreSQL)**

### Фреймворки / бібліотеки

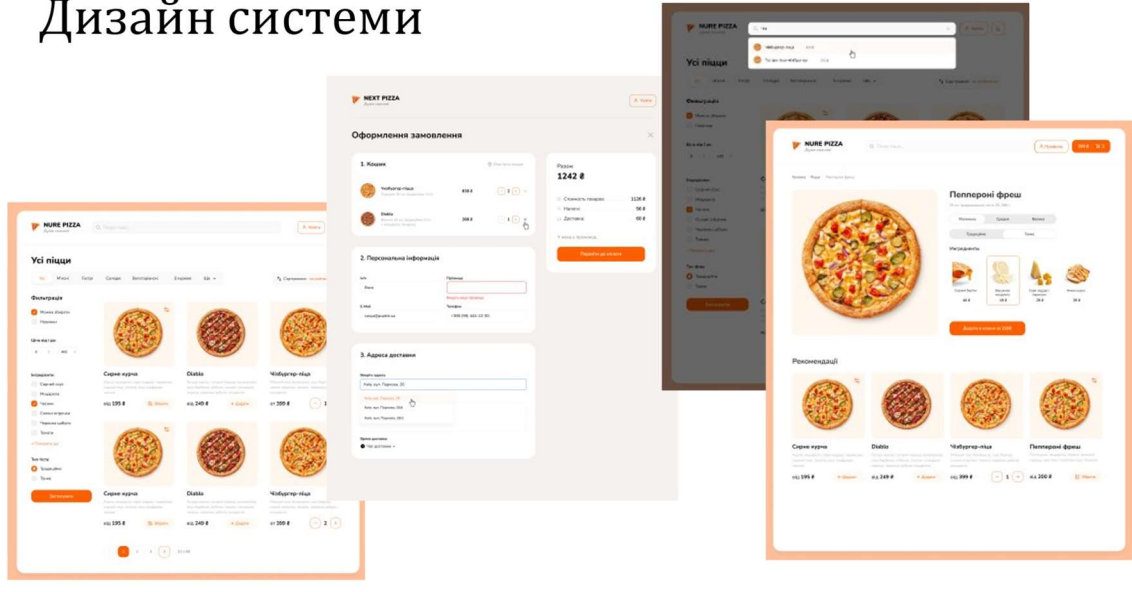
- Next.js • React** — UI + SSR
- Tailwind CSS** — швидка стилізація
- Prisma ORM** — типізовані запити
- NextAuth** — Google OAuth + email
- Zustand** — глобальний стан
- Node.js** — серверне оточення
- LiqPay SDK** — онлайн-оплата
- Google Maps API** — адреси / гео
- Збереження на Vercel.**



Діаграма розгортання веб-сервісу піцерії

Рисунок Н.8 – Слайд 8

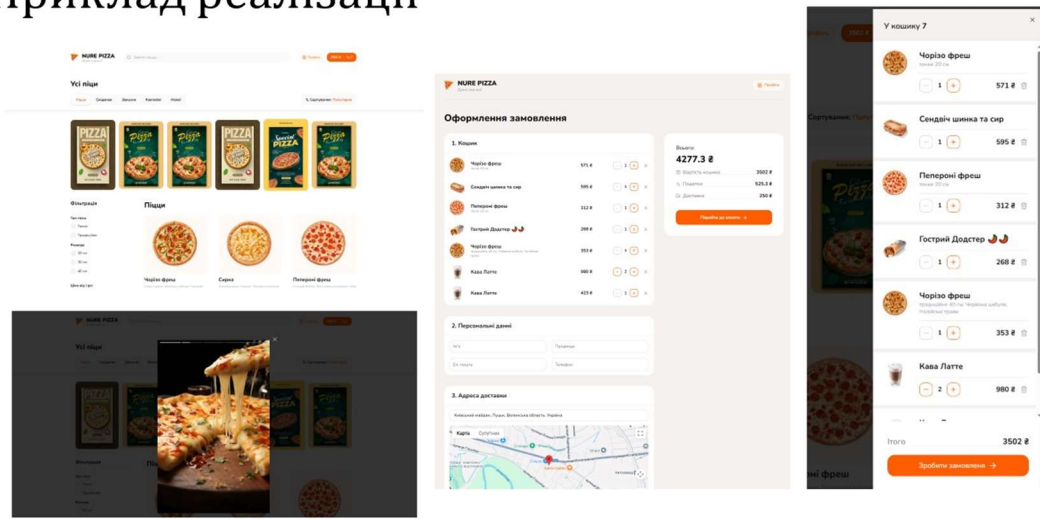
# Дизайн системи



9

Рисунок Н.9 – Слайд 9

# Приклад реалізації



10

Рисунок Н.10 – Слайд 10

## Фрагменти коду з цікавими моментами розробленого програмного забезпечення

```

interface Props {
  description: string;
  orderId: number;
  amount: number;
}
export async function createLiqPayPayment(details: Props) {
  const public_key = process.env.LIQPAY_PUBLIC_KEY!;
  const private_key = process.env.LIQPAY_PRIVATE_KEY!;
  const params = {
    version: '3',
    public_key,
    action: 'pay',
    amount: details.amount.toString(),
    currency: 'UAH',
    description: details.description,
    order_id: details.orderId.toString(),
    language: 'ua',
    result_url: process.env.LIQPAY_RESULT_URL,
    server_url: process.env.LIQPAY_CALLBACK_URL,
  };

  const data = Buffer.from(JSON.stringify(params)).toString('base64');

  const signature = crypto
    .createHash('sha1')
    .update(private_key + data + private_key)
    .digest('base64');

  const paymentUrl = `https://www.liqpay.ua/api/3/checkout?data=${data}&signature=${signature}`;

  return { data, signature, paymentUrl };
};

```

11

Рисунок Н.11 – Слайд 11

## Фрагменти коду з цікавими моментами розробленого програмного забезпечення

The image displays three screenshots of the LIQPAY mobile application interface:

- Payment Screen:** Shows a QR code for payment, the amount of 421.00 UAH, and a 'Pay' button. Below the button are options for 'Pay' and 'Cancel'.
- Confirmation Screen:** Displays a green checkmark and the text 'Дякуємо! Платіж успішно відправлено.' (Thank you! Payment successfully sent). It includes details: ID: 2663336596, Date: 17.06.2025 18:00:47, and Sum: 421.00 UAH. A button 'Завантажити квитанцію' (Download receipt) is visible.
- Receipt Screen:** Shows a receipt for order 'Замовлення №1' with a total of 421.00 UAH. It lists payment details: Order ID: 2663336596, Date: 17.06.2025 18:00:47, and Sum: 421.00 UAH. It also includes a QR code for the receipt and a 'Завантажити квитанцію' button.

12

Рисунок Н.12 – Слайд 12

# Тестування

```
PS C:\dev\next-pizza> node testPerformance.js
Час виконання Prisma-запиту: 31 ms
{
  "id": 20,
  "name": "Чоризо фреш",
  "imageUrl": "https://media.dodostatic.net/image/r:584x584/11EE7D61786D472F9A5D71E894149384.webp",
  "categoryId": 1,
  "createdAt": "2025-01-18T08:10:58.986Z",
  "updatedAt": "2025-01-18T08:10:58.986Z",
  "ingredients": [
    {
      "id": 16,
      "name": "Сир Моцарела",
      "price": 79,
      "imageUrl": "https://cdn.dodostatic.net/static/Img/Ingredients/00003A39D824A82E11E9AFA680FFC349",
      "createdAt": "2025-01-18T08:10:57.812Z",
      "updatedAt": "2025-03-04T21:12:26.948Z"
    }
  ]
}
```

Результат тестування фільтрації

Середній результат на локальній машині - 42 мс при 10 повтореннях, що підтверджує коректність індексів і прийнятний час відповіді.



13

Рисунок Н.13 – Слайд 13

# Публікація результатів



УДК 004.04-502.174

**Навіщо піперії потрібен веб-сайт і як він допомагає клієнтам?**

Мартинів Б.В.  
Науковий керівник – доц. каф. ПІ Гурало І.В.  
Харківський національний університет радіоелектроніки, каф. ПІ, м.  
Харків, Україна  
тел. +38(095) 449-22-09, e-mail: [bohdan.martynov1@nure.ua](mailto:bohdan.martynov1@nure.ua)

Останнім часом сегмент ринку громадського харчування активно розвивається. Останніми роками в Україні все більше вкорінюється звичка харчуватися поза домом. Це пов'язано зі зростанням доходу населення, більш швидким темпом життя і відсутністю часу на готування.

За прогнозами, попитом користуватимуться заклади з якісною, але водночас недорогою їжею і швидким обслуговуванням.

Піперії є невід'ємною частиною сучасної культури харчування і мають безліч переваг. Вони пропонують широкий вибір піц, швидко і зручне обслуговування, комфортну атмосферу, смачне і корисне харчування, а також сприяють розвитку економіки. Тому не дивно, що піперії так популярні й відвідуються людьми різного віку та соціальних верств.

На ринок виходять ті піперії, які оптимізують і спрощують зв'язок між бізнесом і клієнтом. Web-сайт підвищує рівень довіри і показує потенційним споживачам відкритість і готовність піперії до взаємодії.

На теми розвитку піперії впливають інновації в галузі технологій сервісу обслуговування і доставки.

Розвиток нових інформаційних технологій і загальна комп'ютеризація привели до того, що будь-який користувач Інтернету має змогу, не виходячи з дому, отримати повну інформацію на цікаву для нього тему, наприклад, інформацію про заклади свого міста. На сьогодні практично кожна організація має власний веб-сайт.

Недостатньо просто зробити візитку в інтернеті. Ресурс має бути професійним, із сучасним дизайном і зрозумілим інтерфейсом. Клієнту під час відвідування веб-сайту має бути максимально зрозуміло, що він може зробити далі - переглянути пропонувані продукти, уточнити ціакуву для нього інформацію через форму зворотного зв'язку або зробити замовлення.

Для піперії з наявністю веб-сайту з'являється можливість надавати своїм клієнтам більше інформації про заклад, тим самим вказувати клієнту більше приводів прийти саме в цю піперію. За відсутності веб-сайту, щоб клієнт міг вирішити, чи варто йти в будь-який заклад, йому потрібно буде прийти в це місце, подивитися піцу, інтер'єр, дізнатися всі необхідні дані, взяти візитку. Із

14

Рисунок Н.14 – Слайд 14

## Підсумки

### Реалістичність та корисність отриманих результатів

Створений веб-сервіс уже функціонує в тестовій піцерії та щодня приймає понад сотню онлайн-замовлень. Інтеграція з LiqPay і Google Auth забезпечує безпечну оплату й швидку авторизацію, а кастомізація піци під потреби клієнта підвищує середній чек і лояльність. Навантажувальне тестування підтвердило стабільність платформи та час оформлення замовлення < 30 с.

### Можливості використання

Рішення легко запровадити в малих і середніх закладах громадського харчування — достатньо підписки або моделі «pay-as-you-go». Власник отримує брендований сайт, панель управління меню та аналітику продажів, а користувач — мобільний PWA-інтерфейс і миттєві push-сповіщення.

### Можливий розвиток програмного забезпечення

У дорожній карті — live-трекінг доставки через Google Maps, маркет-плейс напоїв і десертів, а також модуль AI-рекомендацій для персональних пропозицій. Наступний етап — випуск нативних мобільних додатків та інтеграція з Apple Pay / Google Pay для ще швидших платежів.



## ДОДАТОК П

## Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ

Дата звіту 6/9/2025

Дата редагування —

Звіт не був оцінений

## Звіт подібності

---

### метадані

Назва організації  
**Kharkiv National University of Radio Electronics**

Заголовок  
**2025\_Б\_ПІ\_ПЗПІ\_21\_7\_Мартинов\_Б\_В**

Автор Науковий керівник / Експерт  
**Мартинов Богдан Валерійовичдоц. Груздо І.В./Нечволод В.Ю.**

підрозділ  
**каф. ПІ**

### Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.

**1.62%**  
1.62% КП 1

**0.55%**  
0.55% КЦ

**25**

Довжина фраз для коефіцієнта подібності 2

**8027**

Кількість слів

**62694**

Кількість символів

### Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навісний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		1
Інтервали		0
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)		11

### Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

#### 10 найдовших фраз Колір тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	<a href="https://software.nure.ua/wp-content/uploads/2024/01/my_2023-end.pdf">https://software.nure.ua/wp-content/uploads/2024/01/my_2023-end.pdf</a>	48 0.60 %
2	<a href="https://software.nure.ua/wp-content/uploads/2024/01/dyplom-mag-7_dotm_docx">https://software.nure.ua/wp-content/uploads/2024/01/dyplom-mag-7_dotm_docx</a>	17 0.21 %
3	<a href="https://openarchive.nure.ua/bitstreams/a781f5af-aec2-491c-b20d-90e27885e1d4/download">https://openarchive.nure.ua/bitstreams/a781f5af-aec2-491c-b20d-90e27885e1d4/download</a>	15 0.19 %

## ДОДАТОК Р

## Перелік джерел посилання за науковими напрямками керівника та науковців кафедри програмної інженерії

26. Методичні вказівки до лабораторних робіт з дисципліни «Емпіричні методи програмної інженерії» для студентів усіх форм навчання першого (бакалаврського) рівня вищої освіти спеціальності 121 – Інженерія програмного забезпечення, освітня програма Програмна інженерія / Упоряд.: І.В. Груздо, О.С. Назаров. – Електронне видання. – Харків: ХНУРЕ, 2024. – 127 с.

27. Методичні вказівки до практичних робіт з дисципліни «Емпіричні методи програмної інженерії» для студентів усіх форм навчання першого (бакалаврського) рівня вищої освіти спеціальності 121 – Інженерія програмного забезпечення, освітня програма Програмна інженерія / Упоряд.: І.В. Груздо, О.С. Назаров. – Харків: ХНУРЕ, 2021. – 80 с.

28. Методичні вказівки до лабораторних робіт з дисципліни «Юзабіліті» для студентів усіх форм навчання першого (бакалаврського) рівня вищої освіти спеціальності 121 – Інженерія програмного забезпечення, освітня програма Програмна інженерія / Упоряд.: І.В. Груздо, Д.О. Колесников, А.О. Функендорф. – Харків: ХНУРЕ, 2020. – 76 с.

29. «Системні технології» 4 (153) 2024 «System technologies», с. 47-57  
DOI: 10.34185/1562-9945-4-153-2024-06

30. Пироженко, С. С. Дослідження продуктивності методів та засобів контролю глобального стану компонентів React додатків / С. С. Пироженко, Н. С. Лесна // Наука онлайн : міжнародний електронний науковий журнал. – 2020. – №10. – С. 13. – URL: <https://nauka-online.com/publications/informatsionnye-tehnologii/2020/10/doslidzhennya-produktivnosti-metodiv-ta-zasobiv-kontrolyu-globalnogo-stanu-komponentiv-react-dodatki/>.

31. Соловей, І. В. Розробка програмної системи для ведення цифрової картки домашньої тварини / І. В. Соловей // Системні технології. – 2024. – № 4

(153). – С. 47–57. – DOI: 10.34185/1562-9945-4-153-2024-06.

32. Методичні вказівки до курсового проектування з дисципліни «Бази даних» для студентів усіх форм навчання спеціальності 121 – «Інженерія програмного забезпечення» (освітньо-професійна програма «Програмна інженерія») / упоряд.: О. О. Мазурова, М. С. Широкопетлева, Ю. Ю. Черепанова, Д. О. Колесников. – Харків : ХНУРЕ, 2020. – 52 с. – URL: <https://catalogue.nure.ua/download=237424>.

## ДОДАТОК С

### Тези з міжнародної науково-технічної конференції

УДК 004.04:502.174

#### **Навіщо піцерії потрібен web-сайт і як він допомагає клієнтам?**

Мартинов Б.В.

Науковий керівник – доц. каф. ПІ Груздо І.В.

Харківський національний університет радіоелектроніки, каф. ПІ, м.

Харків, Україна

тел. +38(095) 449-22-09, e-mail: [bohdan.martynov1@nure.ua](mailto:bohdan.martynov1@nure.ua)

Останнім часом сегмент ринку громадського харчування активно розвивається. Останніми роками в Україні все більше вкорінюється звичка харчуватися поза домом. Це пов'язано зі зростанням доходу населення, більш швидким темпом життя і відсутністю часу на готування.

За прогнозами, попитом користуватимуться заклади з якісною, але водночас недорогою їжею і швидким обслуговуванням.

Піцерії є невід'ємною частиною сучасної культури харчування і мають безліч переваг. Вони пропонують широкий вибір піц, швидко і зручне обслуговування, комфортну атмосферу, смачне і корисне харчування, а також сприяють розвитку економіки. Тому не дивно, що піцерії так популярні й відвідуються людьми різного віку та соціальних верств.

На ринок виходять ті піцерії, які оптимізують і спрощують зв'язок між бізнесом і клієнтом. Web-сайт підвищує рівень довіри і показує потенційним споживачам відкритість і готовність піцерії до взаємодії.

На темпи розвитку піцерії впливають інновації в галузі технологій сервісу обслуговування і доставки.

Розвиток нових інформаційних технологій і загальна комп'ютеризація призвели до того, що будь-який користувач Інтернету має змогу, не виходячи з дому, отримати повну інформацію на цікаву для нього тему, наприклад, інформацію про заклади свого міста. На сьогодні практично кожна організація має власний web-сайт.

Недостатньо просто зробити візитну картку в інтернеті. Ресурс має бути професійним, із сучасним дизайном і зрозумілим інтерфейсом. Клієнту під час відвідування web-сайту має бути максимально зрозуміло, що він може зробити далі - переглянути пропонований продукт, уточнити цікаву для нього інформацію через форму зворотного зв'язку або зробити замовлення.

Для піцерії з наявністю web-сайту з'являється можливість надавати своїм клієнтам більше інформації про заклад, тим самим вказувати клієнту більше приводів прийти саме в цю піцерію. За відсутності web-сайту, щоб клієнт міг вирішити, чи варто йти в будь-який заклад, йому потрібно буде прийти в це місце, подивитися ціни, інтер'єр, дізнатися всі необхідні дані, взяти візитку. Із

сучасним темпом життя мало хто на це стане витратити свій час. Швидше за все, людина піде в той заклад, який вже знає або подивиться web-сайти інших піцерій і зробить свій вибір.

Маючи web-сайт, можливо показати всім клієнтам переваги саме вашого закладу, схему проїзду, контактні дані, години роботи. Web-сайт дасть змогу заздалегідь повідомити клієнту про пропозиції, про новинки, акції, бонуси і навіть надасть можливість віддалено оформити замовлення. На web-сайті можна створити каталог продукції з можливістю вибору інгредієнтів, сформувавши заздалегідь замовлення, щоб клієнт витрачав на оформлення замовлення мінімум зусиль і часу.

Web-сайт дає змогу спростити процес замовлення. На web-сайті можна запропонувати відвідувачам можливість онлайн-оплати і відстеження статусу замовлення. Також можна розглянути можливість оформлення замовлення з доставкою. Web-сайт може дозволити дізнатися час доставки. Залежно від адреси доставки і часу готовності страви, система в режимі онлайн розрахує, о котрій годині замовлення буде доставлено клієнту, з функцією інформування.

Ну і звичайно, з наявністю web-сайту з'являється можливість клієнтам рекомендувати вашу піцерію іншим. З web-сайтом це зробити набагато простіше - достатньо лише вислати посилання на web-сайт. Людині не доведеться пояснювати на словах, що за піцерія, які там ціни і де вона розташована. Ви підвищуєте престиж закладу. Очевидно, що для нового клієнта, піцерія, у якої є web-сайт, сприймається більш сучасним, ніж така ж піцерія без web-сайту.

З наявністю web-сайту з'являється більше можливостей для вдосконалення. Можливо підключити на web-сайті будь-який плагін. Наприклад, сервіс замовлення столиків із прив'язкою до sms-системи. Або кнопку онлайн-дзвінка, онлайн-чату, щоб чіпляти клієнта швидко, тут же зв'язатися телефоном, щоб замовити піцу, столик. Ви можете підключити інтернет-магазин, блог, спливаючі рекламні оголошення.

Таким чином, у кожній піцерії обов'язково має бути web-сайт.

У постійно мінливих умовах сильної конкуренції на ринку громадського харчування необхідно розробляти ефективні маркетингові стратегії для залучення й утримання клієнтів, а наявність власного web-сайту для піцерії в умовах використання сучасних інформаційних технологій - це необхідний чинник існування, що дає змогу розширити поле рекламної діяльності та залучити тим самим додаткових клієнтів.

## Список використаних джерел:

Богданова Л.В. Маркетингові інструменти просування ресторанного бізнесу в Інтернеті // Маркетинг і цифрові технології. – 2020. – №3. – С. 45-54.

Ковальчук Н.І. Роль сайту і соціальних мереж у просуванні бренду ресторанних закладів // Вісник Київського національного торговельно-економічного університету. – 2021. – №1. – С. 75–82.

Мельник О.В. Впровадження інформаційних технологій у ресторанному бізнесі: електронні замовлення та управління ресурсами // Економіка та управління підприємством. – 2019. – №2. – С. 60–66.

Johnson, M. How Restaurant Websites Improve Consumer Engagement // Journal of Hospitality and Tourism Technology. – 2019. – Vol. 10, No. 2. – P. 145-158.

Sethi, P. The Importance of a Restaurant Website for Customer Engagement // Journal of Hospitality & Tourism Marketing. – 2018. – Vol. 29, No. 4. – P. 312-327.

