

**Вищий навчальний заклад  
«Університет економіки та права «КРОК»  
Фаховий коледж**

Циклова комісія з інформаційних технологій

**Кваліфікаційна робота фахового молодшого  
бакалавра**

на тему Розробка інтернет-магазину піци з індивідуальними рецептами та  
авторськими смаками

Виконав \_\_\_\_\_  
(Підпис)  
Мартинов Данііл Сергійович  
(прізвище, ім'я, по батькові)

Науковий керівник \_\_\_\_\_  
Пантєєв Роман Леонідович  
(прізвище, ім'я, по батькові)

(Резолюція «До захисту»)

**Попередній захист:**

\_\_\_\_\_  
(Висновок: “До захисту в екзаменаційній комісії”)

**Голова циклової комісії**

\_\_\_\_\_  
(Підпис )

\_\_\_\_\_  
(Прізвище, ініціали)

\_\_\_\_\_  
(Дата)

**Київ – 2025 року**

**ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД**  
**УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»**

Фаховий коледж

**Циклова комісія з інформаційних технологій**

Спеціальність 121 інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Голова циклової комісії \_\_\_\_\_ Леонід УВАРОВ

(підпис)

« \_\_\_\_ » \_\_\_\_\_ 2025 року

**ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Здобувач освіти Мартинов Данііл Сергійович

1. Тема роботи Розробка інтернет-магазину піци з індивідуальними рецептами та авторськими смаками затверджена наказом по університету

від « \_\_\_\_ » \_\_\_\_\_ 202\_\_ р. № \_\_\_\_\_

2. Термін здачі закінченої роботи «30» травня 2025 року

**3. Вихідні дані до роботи:**

- 1) **цільова аудиторія** – користувачі онлайн-сервісів доставки їжі, молодь, сім'ї, офісні працівники, люди з дієтичними обмеженнями (вегани, безглютенові, тощо).
- 2) **функціональність** – творення індивідуальної піци, перегляд авторських рецептів, оформлення та оплата замовлення, історія покупок, відгуки, сповіщення, система лояльності.
- 3) **технічні вимоги** - вебплатформа, мови програмування JavaScript/Python, фреймворки React/Vue.js, Node.js/Django, база даних PostgreSQL/MongoDB, інтеграція з платіжними та поштовими сервісами.
- 4) **можливість інтеграції (взаємодії)** - CRM-системи, служби доставки (Glovo, Raketa), платіжні шлюзи (LiqPay, Stripe), Telegram-боти, email/SMS сервіси.
- 5) **існуючі рішення та кращі практики у сфері розробки** - Domino's, Papa John's, Mister Cat; використання конструктора піци, адаптивного дизайну, швидкого UX, персоналізації замовлень.

#### 4. Зміст пояснювальної записки

- 1) **Розділ 1 Теоретична частина.** Розглянуто сучасні сервіси доставки їжі та їх функціонал персоналізації. Проведено огляд технологій створення інтернет-магазинів з адаптивним дизайном. Визначено кращі практики реалізації конструкторів піци. Обґрунтовано вибір стеку технологій для проекту.
- 2) **Розділ 2 Проєктування та розробка.** Описано структуру інтерфейсу, логіку обробки замовлень та формування ціни. Визначено архітектуру клієнтської й серверної частин. Реалізовано інтеграцію з базою даних, платіжними сервісами, доставкою і Telegram-ботом. Сформовано бізнес-логіку застосунку
- 3) **Розділ 3 Експериментальна частина.** Проведено тестування сайту на різних пристроях і виправлення помилок. Здійснено оцінку відповідності функціоналу очікуванням користувачів. Надано інструкції для користувачів і адміністраторів. Розглянуто аспекти захисту персональних даних.
- 4) **Розділ 4. Реалізація інтернет-магазину піци.** Описано реалізацію вебзастосунку з використанням React, Node.js та PostgreSQL. Реалізовано кастомний конструктор піци, реєстрацію, замовлення й оплату. Інтегровано зовнішні сервіси: доставка, оплата, Telegram-бот. Оцінено ефективність та можливості масштабування.

#### 5. Перелік графічного матеріалу:

- Структура бази даних інтернет-магазину піци
- Код клієнтської частини (Frontend)
- Код серверної частини (Backend)
- Скріншоти розробленого сайту

Дата видачі завдання «12» лютого 2025 року

Науковий керівник

\_\_\_\_\_

(підпис)

Пантєєв Р. Л.

( прізвище, ім'я, по батькові)

Завдання прийняла до виконання

\_\_\_\_\_

(підпис)

Мартинів Д. С.

( прізвище, ім'я, по батькові)

## РЕФЕРАТ

**Пояснювальна записка:** 60 сторінка, 11 рисунків, 7 таблиць, 6 додатків, 15 джерел.  
**Об'єкт дослідження** — вебзастосунок інтернет-магазину піци з підтримкою кастомізації замовлень.

**Мета роботи** — розробка програмного забезпечення для онлайн-магазину піци з можливістю створення індивідуальних рецептів, замовлення через адаптивний інтерфейс, інтеграцією з платіжними системами, службами доставки та месенджером Telegram.

У кваліфікаційній роботі представлено результати розробки вебзастосунку з реалізацією функціоналу конструктора піци, авторизації користувачів, формування замовлень, розрахунку вартості та оплати. Проведено аналіз ринку онлайн-доставки їжі, описано архітектуру клієнтської та серверної частин системи. Вебінтерфейс реалізовано з використанням HTML, CSS (Tailwind) і JavaScript (React), серверну частину — на Node.js з Express, база даних — PostgreSQL. Розроблено REST API для взаємодії компонентів, здійснено інтеграцію з зовнішніми сервісами. Тестування охопило функціональність, юзабіліті та навантаження.

Результати можуть бути застосовані в комерційних онлайн-сервісах доставки їжі.

**Ключові слова:** ІНТЕРНЕТ-МАГАЗИН, КОНСТРУКТОР ПІЦИ, ВЕБЗАСТОСУНОК, РЕАКТИВНИЙ ІНТЕРФЕЙС, NODE.JS, POSTGRESQL.

## ABSTRACT

**Explanatory note:** 60 pages, 11 figures, 7 tables, 6 appendices, 15 sources.  
**The object of the study** is a web application for an online pizza store with customizable orders.  
**The purpose of the work** is to develop a software solution for a pizza e-commerce platform that enables users to create personalized recipes, place orders via an adaptive interface, and integrate with payment, delivery, and Telegram messaging services.

The qualification work presents the development of a pizza web application featuring a pizza constructor, user authentication, price calculation, ordering and payment functionality. Market analysis of food delivery services was conducted. The system architecture is described: frontend is built using HTML, Tailwind CSS, and React; backend — using Node.js with Express; database — PostgreSQL. A REST API was developed for component communication. The system is integrated with external services and underwent usability, performance, and functional testing.

The solution can be implemented in commercial food delivery platforms.

**Keywords:** ONLINE STORE, PIZZA CONSTRUCTOR, WEB APPLICATION, REACTIVE INTERFACE, NODE.JS, POSTGRESQL.

## ЗМІСТ

|   |    |
|---|----|
| СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ .....                          | 6  |
| ВСТУП.....  | 7  |
| РОЗДІЛ 1. ЗАГАЛЬНІ ПОЛОЖЕННЯ.....                           | 9  |
| 1.1. Актуальність теми .....                                | 9  |
| 1.2. Мета і завдання дослідження.....                       | 9  |
| 1.3. Об’єкт і предмет дослідження .....                     | 10 |
| 1.4. Методи дослідження .....                               | 11 |
| РОЗДІЛ 2. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....                   | 13 |
| 2.1. Сучасний стан ринку онлайн-замовлень піци .....        | 13 |
| 2.2. Аналіз функціоналу конкурентів .....                   | 14 |
| 2.3. Вивчення потреб цільової аудиторії .....               | 15 |
| 2.4. Унікальність авторських рецептів та кастомізації ..... | 17 |
| 2.5. Визначення функціональних вимог до системи .....       | 18 |
| РОЗДІЛ 3. ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....           | 21 |
| 3.1. Вибір архітектури вебзастосунку .....                  | 21 |
| 3.2. Технологічний стек для реалізації.....                 | 21 |
| 3.3. Моделювання бази даних .....                           | 22 |
| 3.4. Розробка структури інтерфейсу користувача .....        | 23 |
| 3.5. Прототипування конструктора піци.....                  | 24 |
| 3.6. Безпека та захист персональних даних .....             | 24 |
| РОЗДІЛ 4. РЕАЛІЗАЦІЯ ІНТЕРНЕТ-МАГАЗИНУ ПІЦИ.....            | 26 |
| 4.1. Створення клієнтської частини (Frontend).....          | 26 |
| 4.2. Розробка серверної частини (Backend) .....             | 26 |
| 4.3. Реалізація конструктора піци .....                     | 27 |
| 4.4. Додавання та управління авторськими рецептами.....     | 27 |
| 4.5. Система обробки замовлень .....                        | 28 |
| 4.6. Тестування основних функцій системи .....              | 28 |
| ВИСНОВКИ .....  | 30 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....                            | 33 |
| ДОДАТКИ .....   | 35 |
| Додаток А. Код клієнтської частини (Frontend) .....         | 35 |
| Додаток Б. Код серверної частини (Backend).....             | 52 |
| Додаток В. Скріншоти розробленого сайту .....               | 56 |

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

**API** — Application Programming Interface — інтерфейс програмування додатків

**UI** — User Interface — користувацький інтерфейс

**UX** — User Experience — користувацький досвід

**HTML** — HyperText Markup Language — мова розмітки гіпертексту

**CSS** — Cascading Style Sheets — каскадні таблиці стилів

**JS** — JavaScript — мова програмування для клієнтської частини вебзастосунків

**SQL** — Structured Query Language — мова структурованих запитів до бази даних

**DB** — Database — база даних

**REST** — Representational State Transfer — архітектурний стиль для побудови API

**CRUD** — Create, Read, Update, Delete — базові операції з даними

**React** — JavaScript-бібліотека для побудови інтерфейсів користувача

**Node.js** — середовище виконання JavaScript на серверній стороні

**PostgreSQL** — об'єктно-реляційна система керування базами даних

**JSON** — JavaScript Object Notation — формат обміну структурованими даними

**HTTP** — HyperText Transfer Protocol — протокол передавання гіпертексту

**URL** — Uniform Resource Locator — уніфікований локатор ресурсу

## ВСТУП

**Актуальність завдання.** У сучасних умовах стрімкого розвитку цифрових технологій та зростання популярності електронної комерції значну роль відіграють онлайн-сервіси з індивідуалізованим підходом до клієнта. Однією з найбільш затребуваних сфер у цьому напрямку є онлайн-замовлення їжі. Особливою популярністю користується піца — страва, що легко піддається варіаціям за складом і смаком. Більшість існуючих сервісів пропонують обмежену кількість фіксованих комбінацій, не враховуючи індивідуальні смаки та потреби користувача. Це створює нішу для розвитку інтернет-магазину нового покоління, що дозволяє створювати піцу за власним рецептом, комбінуючи інгредієнти самостійно, а також пропонує унікальні авторські смаки [14][15]. Актуальність теми обумовлена потребою в інноваційних рішеннях для індустрії громадського харчування та зростанням попиту на персоналізовані сервіси [14].

**Мета роботи.** Метою дипломної роботи є розробка функціонального вебзастосунку — інтернет-магазину піци з можливістю індивідуального формування рецепту користувачем і вибору авторських смаків, що забезпечує зручність замовлення, гнучкий вибір інгредієнтів та підтримку онлайн-оплати [1][2].

**Завдання роботи.** Для досягнення мети потрібно виконати наступні завдання:

- провести аналіз існуючих рішень у сфері онлайн-замовлень піци [14][15];
- вивчити потреби цільової аудиторії [15];
- сформулювати функціональні та нефункціональні вимоги до системи [1][2];
- обрати відповідні інструменти розробки [1][2][9][10];
- спроектувати архітектуру системи та базу даних [3][5];
- реалізувати вебінтерфейс з функцією конструктора піци [2][17];
- створити систему додавання авторських рецептів;
- впровадити механізми реєстрації, замовлення та оплати [12];
- провести тестування функціоналу вебзастосунку [13].

**Об'єкт дослідження.** Об'єктом дослідження є процес розробки вебзастосунків для електронної комерції у сфері харчування [1][6].

**Предмет дослідження.** Предметом дослідження є методи та засоби створення інтернет-магазину з підтримкою індивідуалізації товару та інтеграцією з платіжними сервісами [10][12].

**Методи дослідження.** У дипломній роботі використовуються такі методи дослідження:

- аналітичний метод — для вивчення аналогів і потреб ринку [14][15];
- моделювання — для розробки структури системи та бізнес-процесів [3][5];
- метод прототипування — для створення конструктора піци [2][17];
- практичний метод — безпосередня розробка програмного забезпечення [1][9][10];
- експериментальний метод — тестування працездатності системи [13].

**Практичне значення одержаних результатів.** Результатом роботи є повнофункціональний прототип інтернет-магазину піци з інтерфейсом користувача, можливістю кастомізації продукту та адміністрування авторських рецептів. Розроблений застосунок може бути впроваджений у практичну діяльність підприємств харчової галузі та адаптований для різних типів доставки їжі. Система має потенціал до масштабування та подальшого комерційного використання [1][2][14].

## РОЗДІЛ 1. ЗАГАЛЬНІ ПОЛОЖЕННЯ

### 1.1. Актуальність теми

У сучасному цифровому середовищі все більшого значення набувають онлайн-сервіси, що забезпечують швидкий доступ до товарів і послуг. Сфера доставки їжі, зокрема піци, є однією з найдинамічніших у сегменті електронної комерції [14]. Ринок онлайн-замовлень активно розвивається завдяки змінам у стилі життя споживачів, зростанню використання мобільних додатків та популяризації безконтактної доставки [14][15].

Попри велику кількість сервісів доставки, переважна більшість з них пропонує обмежений набір фіксованих варіантів піци, без можливості створення власного рецепту. Це не відповідає сучасному тренду персоналізації, коли споживач прагне мати більше контролю над кінцевим продуктом, особливо у випадках дієтичних обмежень, алергій або індивідуальних смакових вподобань [14][15].

Інтернет-магазин, який дозволяє користувачам формувати власну піцу, обираючи інгредієнти, а також пропонує унікальні авторські рецепти, здатен заповнити цю нішу. Такий підхід підвищує лояльність клієнтів, збільшує середній чек і дозволяє бізнесу виділитися серед конкурентів [15].

Таким чином, розробка інтернет-магазину піци з можливістю індивідуального формування рецепту є актуальним завданням, що відповідає потребам сучасного ринку, а також технологічним можливостям реалізації [1][2].

### 1.2. Мета і завдання дослідження

Метою даної дипломної роботи є розробка інтернет-магазину піци, який надає користувачам можливість створювати індивідуальні рецепти шляхом вибору інгредієнтів, а також обирати з-поміж авторських смаків, інтегруючи сучасні вебтехнології для зручного та безпечного замовлення [1][2].

Цей застосунок має забезпечити інтуїтивно зрозумілий інтерфейс, гнучкість у формуванні замовлень, ефективну обробку інформації про інгредієнти, а також можливість онлайн-оплати та адміністрування замовлень [3][6].

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- провести аналіз ринку онлайн-сервісів доставки піци [14][15];
- дослідити досвід конкурентів та потреби цільової аудиторії [15];
- сформулювати функціональні та нефункціональні вимоги до інформаційної системи [1][2];
- визначити архітектуру та технології, що будуть використані при розробці [3][5];
- спроектувати структуру бази даних та бізнес-логіку системи [5][8];
- реалізувати вебінтерфейс для формування індивідуальних рецептів піци [2][9];
- впровадити функціонал управління авторськими рецептами [2][9];
- створити систему реєстрації, входу та обробки замовлень [10][11];
- реалізувати інтеграцію з платіжним сервісом [12];
- протестувати систему на функціональність, зручність використання та надійність [13].

### **1.3. Об'єкт і предмет дослідження**

Об'єктом дослідження в даній роботі є процес розробки та впровадження вебзастосунків, які призначені для електронної комерції, а саме для сфери доставки їжі через інтернет [1][2]. Сучасні вебзастосунки надають можливість користувачам робити замовлення у зручний спосіб без необхідності фізичної присутності у закладі, що значно підвищує доступність послуг та комфорт користувача [6][14]. У контексті дослідження особливу увагу приділено специфіці інтернет-магазинів з можливістю індивідуалізації продукту — у цьому випадку, створенню піци з унікальним набором інгредієнтів, а також вибору авторських рецептів [2][15].

Вивчення об'єкта дозволяє оцінити поточні тенденції розвитку електронної торгівлі у сфері громадського харчування, виявити існуючі проблеми та можливості для вдосконалення процесів замовлення їжі онлайн [14][15]. Крім того, об'єкт дослідження включає як технічні, так і організаційні

аспекти розробки вебзастосунку, що вимагає комплексного підходу та використання сучасних інформаційних технологій [3][5].

Предметом дослідження є конкретні методи, підходи, технології і засоби розробки інтернет-магазину піци з індивідуальним підходом до формування замовлення [2][9]. Особлива увага приділяється реалізації функціоналу, який дозволяє користувачам обирати та комбінувати інгредієнти для створення власного рецепту піци, а також використовувати авторські рецепти, що підвищує унікальність пропозиції та відповідає сучасним вимогам персоналізації [2][9].

Предмет дослідження також охоплює розробку інтерфейсу користувача, побудову структури бази даних для зберігання інформації про інгредієнти, рецепти та замовлення, а також інтеграцію системи із платіжними сервісами для забезпечення безпечної оплати онлайн [5][8][12]. Важливим аспектом є забезпечення надійності та зручності використання системи, що впливає на загальний рівень задоволеності користувачів та ефективність роботи інтернет-магазину [13].

Таким чином, об'єкт і предмет дослідження визначають рамки та фокус роботи, спрямованої на створення інноваційного вебзастосунку, який задовольняє сучасні потреби ринку та споживачів у сфері замовлення їжі онлайн із можливістю персоналізації продукту [1][2].

#### **1.4. Методи дослідження**

У процесі виконання дипломної роботи застосовувалися різноманітні наукові та практичні методи дослідження, які дозволили комплексно розглянути проблему розробки інтернет-магазину піци з індивідуальними рецептами та авторськими смаками [1][2][3].

Аналіз літературних джерел і ринку — цей метод використовувався для вивчення сучасного стану ринку онлайн-замовлень їжі, зокрема піци, а також для ознайомлення з існуючими інформаційними системами та технологіями, які застосовуються у подібних проєктах [14][15]. Аналіз дозволив виділити основні тенденції, переваги і недоліки конкурентних сервісів, а також сформулювати вимоги до розроблюваної системи [14].

Метод моделювання — застосовувався для побудови структури інформаційної системи, розробки бази даних, бізнес-процесів і архітектури вебзастосунку. За допомогою UML-діаграм та інших моделей було здійснено візуалізацію основних компонентів системи та їх взаємодії, що полегшило процес розробки та тестування [3][5].

Прототипування — цей метод дав змогу створити початкові варіанти інтерфейсу користувача, у тому числі конструктора піци, що дозволило отримати зворотній зв'язок від потенційних користувачів і внести необхідні корективи на ранніх етапах розробки [2][13].

Практичний метод — реалізація програмного продукту на основі вибраних технологій (Frontend, Backend, база даних) з використанням сучасних мов програмування та фреймворків. Цей метод включає написання коду, налагодження функціоналу і впровадження алгоритмів роботи системи [2][9][10].

Експериментальний метод — тестування розробленої системи, що включає функціональне тестування для перевірки відповідності заявленим вимогам, тестування юзабіліті для оцінки зручності використання, а також навантажувальне тестування для оцінки продуктивності та стабільності роботи системи [13].

Метод порівняння — використовувався для зіставлення розробленого продукту з існуючими аналогами, що допомогло визначити його переваги і недоліки, а також перспективи подальшого вдосконалення [14].

Застосування цих методів дозволило забезпечити комплексний підхід до дослідження, що включає як теоретичний аналіз, так і практичну реалізацію, що є важливим для створення якісного та ефективного вебзастосунку [1][2][3].

## РОЗДІЛ 2. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 2.1. Сучасний стан ринку онлайн-замовлень піци

Ринок онлайн-замовлень піци є однією з найбільш динамічно зростаючих і конкурентних галузей у сфері електронної комерції та доставки їжі. За останні роки значно змінилися споживчі звички та поведінка користувачів: люди все частіше віддають перевагу замовленням через інтернет, мобільні додатки та платформи доставки, що забезпечує швидкість, зручність і широкий вибір.

Поява нових технологій, розвиток мобільного інтернету та зростання популярності смартфонів сприяють зростанню популярності онлайн-замовлень. Внаслідок цього багато піцерій і мереж швидкого харчування активно впроваджують власні вебсайти та мобільні застосунки, а також співпрацюють із агрегаторами доставки, такими як Glovo, Uber Eats, Raketa та іншими.

Основною тенденцією сучасного ринку є підвищена увага до персоналізації замовлень. Клієнти хочуть мати змогу не лише вибрати зі стандартного меню, а й створювати піцу з урахуванням власних смакових вподобань, обираючи інгредієнти та розмір порції. Це дозволяє збільшити лояльність споживачів і підвищити середній чек.

Конкурентне середовище на ринку формує вимоги до якості обслуговування, швидкості доставки та зручності інтерфейсу. Багато сервісів пропонують додаткові можливості, такі як відстеження статусу замовлення в режимі реального часу, акції та програми лояльності, що підвищує привабливість інтернет-магазинів піци.

Водночас, ринок має певні виклики, зокрема потребу в інтеграції з платіжними системами, забезпеченні безпеки персональних даних та створенні інтуїтивно зрозумілих інтерфейсів. Також значним є фактор конкуренції з боку як великих мережевих компаній, так і локальних закладів, які намагаються знайти власну нішу через унікальні пропозиції.

Таким чином, сучасний стан ринку онлайн-замовлень піци характеризується високим рівнем конкуренції, розвитком технологій та змінами

у споживчих запитах, що обумовлює необхідність впровадження інноваційних рішень, таких як індивідуалізація рецептів і створення авторських смаків.

## **2.2. Аналіз функціоналу конкурентів**

Аналіз функціоналу конкурентів є важливим етапом при розробці інтернет-магазину піци, оскільки дозволяє визначити сильні та слабкі сторони існуючих рішень, а також виявити можливості для вдосконалення та впровадження унікальних функцій.

На сучасному ринку онлайн-замовлень піци домінують як великі мережі, так і локальні піцерії, які пропонують свої послуги через власні вебсайти та мобільні застосунки, а також за допомогою агрегаторів доставки. Розглянемо ключові функціональні можливості основних конкурентів:

### **1. Стандартне меню і вибір піци**

Більшість сервісів пропонують фіксований набір піц із заздалегідь сформованими інгредієнтами. Користувач може обирати зі стандартного меню різні розміри та варіанти випікання, проте можливості самостійного формування піци обмежені або відсутні.

### **2. Конструктор піци (Customization)**

Деякі популярні платформи надають користувачам можливість формувати піцу, вибираючи окремі інгредієнти: тип тіста, соуси, сир, м'ясні та овочеві добавки. Проте, часто функціонал конструктора є спрощеним, не дозволяє зберігати власні рецепти або рекомендувати авторські варіанти.

### **3. Авторські рецепти і спеціальні пропозиції**

Деякі конкуренти пропонують унікальні авторські смаки, створені шеф-кухарями або на основі популярних трендів. Ці рецепти часто виділяються у меню окремою категорією, що підвищує привабливість замовлень.

### **4. Інтерфейс користувача**

Великі компанії прагнуть зробити інтерфейс максимально простим та інтуїтивним, зручно розташовуючи категорії, фільтри та підказки. Мобільні додатки мають зручну навігацію та швидкий доступ до основних функцій.

### **5. Система замовлень і оплати**

Конкуренти забезпечують різноманітні способи оплати, включаючи онлайн-платежі, оплату при доставці, використання бонусів та промокодів. Важливою функцією є також відстеження статусу замовлення в режимі реального часу.

#### **6. Додаткові сервіси**

Деякі сервіси надають можливість створення профілю користувача з історією замовлень, налаштуваннями алергій, програмами лояльності, а також підтримкою клієнтів через чат або телефон.

**Висновок.** Аналіз показує, що хоча багато конкурентів пропонують стандартний функціонал замовлення піци, існує потенціал для розвитку через глибшу персоналізацію, можливість збереження власних рецептів, інтеграцію авторських смаків та покращення користувацького досвіду. Це створює основу для розробки інтернет-магазину з розширеним функціоналом, що відповідатиме сучасним вимогам ринку і запитам користувачів.

### **2.3. Вивчення потреб цільової аудиторії**

Для успішної розробки інтернет-магазину піци з індивідуальними рецептами та авторськими смаками важливим є глибоке розуміння потреб і вподобань цільової аудиторії. Цільова аудиторія складається з користувачів, які активно замовляють їжу онлайн, цінують швидкість обслуговування, комфорт та можливість персоналізації продукту.

Основними категоріями користувачів можна виділити:

- **Молодь та студенти (18-30 років):**

Ця група цінує швидкість замовлення, доступність через мобільні пристрої та можливість експериментувати зі смаками. Вони часто користуються акціями та програмами лояльності, активно шукають нові смаки та можливості створити власний рецепт піци.

- **Сім'ї з дітьми:**

Для цієї аудиторії важливі зручність, можливість замовити різноманітне меню для різних членів сім'ї, включаючи опції для дітей.

Часто потрібні алерген-інформація та можливість створення піци з урахуванням дієтичних обмежень.

- **Працюючі професіонали:**

Часто обмежені у часі, ці користувачі шукають швидкі та якісні рішення для обіду або вечері. Для них важливі простота замовлення, можливість швидкої оплати та доставка у визначений час.

- **Любителі здорового харчування та вегетаріанці:**

Ця категорія потребує детальної інформації про склад продуктів, а також можливість створення піци з корисних інгредієнтів без м'яса або з низьким вмістом калорій.

**Основні потреби цільової аудиторії:**

- **Індивідуалізація замовлення:** можливість створювати піцу з урахуванням власних смакових вподобань і дієтичних потреб.

- **Зручність інтерфейсу:** простий і зрозумілий процес замовлення без зайвих кроків.

- **Швидкість обслуговування:** оперативна обробка замовлення та швидка доставка.

- **Надійність оплати:** підтримка різних способів оплати, включаючи онлайн.

- **Інформаційна підтримка:** наявність детальної інформації про інгредієнти, алергени та поживні властивості.

- **Персональні пропозиції:** акції, знижки та рекомендації на основі історії замовлень.

**Методи вивчення потреб:**

Для збору інформації про потреби цільової аудиторії використовувалися такі методи:

- опитування потенційних користувачів через онлайн-анкетування;
- інтерв'ю з представниками різних вікових груп та соціальних категорій;

- аналіз відгуків і коментарів у соціальних мережах та на сайтах конкурентів;
- вивчення аналітики популярних платформ доставки.

#### 2.4. Унікальність авторських рецептів та кастомізації

В умовах високої конкуренції на ринку онлайн-замовлень їжі здатність запропонувати унікальний продукт є ключовою перевагою для інтернет-магазину піци. Саме тому розробка авторських рецептів і можливість індивідуального формування піци (кастомізація) набувають особливого значення.

**Авторські рецепти.** Авторські рецепти — це оригінальні комбінації інгредієнтів, створені шеф-кухарями або командою розробників з урахуванням гастрономічних тенденцій та побажань клієнтів. Вони мають низку переваг:

- **Відмінність від стандартного меню:** Авторські піци виділяються на фоні типових позицій, привертаючи увагу клієнтів, які шукають нові смаки.
- **Емоційний зв'язок з брендом:** Унікальні рецепти підкреслюють індивідуальність компанії, сприяють формуванню лояльності та повторним замовленням.
- **Маркетинговий потенціал:** Авторські рецепти можуть використовуватись у рекламних кампаніях, акціях, сезонних пропозиціях, що стимулює збільшення продажів.

**Кастомізація (Індивідуальне формування піци).** Кастомізація є процесом, що дозволяє користувачам створювати піцу відповідно до власних смакових уподобань шляхом вибору типу тіста, соусу, сиру, основних та додаткових інгредієнтів. Основні переваги кастомізації:

- **Персоналізація замовлення:** Кожен клієнт отримує продукт, максимально адаптований під свої побажання, що підвищує задоволеність і повторюваність замовлень.
- **Гнучкість у виборі:** Можливість виключати інгредієнти, що не подобаються або викликають алергію, а також додавати додаткові компоненти.

- **Розширення асортименту:** За рахунок комбінації інгредієнтів створюється необмежена кількість варіантів піци без необхідності розширювати стандартне меню.

**Поєднання авторських рецептів і кастомізації.** Інтеграція авторських рецептів із можливістю кастомізації надає унікальну пропозицію для клієнтів: користувач може не лише замовити фірмову піцу, а й модифікувати її під власний смак. Це створює додаткову цінність і дозволяє ефективно конкурувати на ринку.

**Висновок.** Унікальність авторських рецептів і гнучкість кастомізації є важливими складовими успішного інтернет-магазину піци. Вони відповідають сучасним трендам персоналізації, сприяють підвищенню лояльності клієнтів і забезпечують конкурентні переваги, що є ключовими для сталого розвитку бізнесу.

## **2.5. Визначення функціональних вимог до системи**

Функціональні вимоги визначають, які конкретні задачі та функції повинна виконувати система, щоб задовольнити потреби користувачів і бізнесу. Для інтернет-магазину піци з індивідуальними рецептами та авторськими смаками визначення чітких функціональних вимог є необхідною умовою успішної розробки та впровадження системи.

### **Основні функціональні вимоги:**

#### **1. Реєстрація та аутентифікація користувачів**

Система повинна підтримувати створення особистого кабінету користувача з можливістю входу за допомогою логіна і пароля, а також авторизацію через соціальні мережі. Особистий кабінет дозволяє зберігати історію замовлень, уподобання та налаштування.

#### **2. Каталог продукції**

Інтернет-магазин повинен мати структурований каталог піци, що включає стандартні позиції, авторські рецепти та можливість створення власної піци. Кожен товар має містити опис, фото, ціну та інформацію про склад інгредієнтів.

### 3. **Конструктор піци (кастомізація)**

Користувач повинен мати можливість вибирати інгредієнти, розмір, тип тіста, соус і інші параметри для створення унікального замовлення. Функція збереження власних рецептів буде перевагою.

### 4. **Кошик замовлень**

Система повинна дозволяти додавати вибрані піци та інші товари до кошика, редагувати кількість, переглядати загальну суму замовлення, а також видаляти позиції.

### 5. **Оформлення замовлення**

Процес оформлення повинен бути зручним і зрозумілим, включати вибір способу доставки, адреси, контактної інформації, а також способів оплати.

### 6. **Оплата онлайн**

Підтримка різних методів оплати: банківські картки, електронні гаманці, оплата при отриманні. Забезпечення безпеки фінансових транзакцій.

### 7. **Відстеження статусу замовлення**

Користувач повинен мати можливість в режимі реального часу бачити статус свого замовлення: прийняте, готується, доставляється, виконане.

### 8. **Адміністративна панель**

Для управління каталогом, обробки замовлень, налаштування акцій і роботи зі звітами повинна бути створена зручна панель адміністратора.

### 9. **Система відгуків і рейтингів**

Користувачі мають можливість залишати відгуки про піцу та сервіс, ставити оцінки, що допомагає підвищувати якість обслуговування.

#### **Додаткові функціональні вимоги:**

- Підтримка багатомовності інтерфейсу (за потребою).
- Налаштування персональних пропозицій, акцій та знижок для користувачів.

- Інформаційна підтримка про алергени та поживну цінність продуктів.

**Висновок.** Визначення функціональних вимог є фундаментом для розробки якісного інтернет-магазину піци, що враховує потреби користувачів і бізнесу. Чітке формулювання цих вимог дозволить оптимізувати процес розробки, забезпечити ефективну роботу системи і високу задоволеність клієнтів.

## РОЗДІЛ 3. ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 3.1. Вибір архітектури вебзастосунку

Архітектура вебзастосунку визначає структуру системи, взаємодію її компонентів та принципи обробки даних. Для інтернет-магазину піци з індивідуальними рецептами важливо обрати таку архітектуру, яка забезпечить високу масштабованість, продуктивність, зручність підтримки і безпеку.

Найбільш поширеним і сучасним підходом є використання **архітектури клієнт-сервер** із розділенням фронтенд- та бекенд-частин. Фронтенд відповідає за інтерфейс користувача та обробку взаємодій, бекенд — за логіку бізнес-процесів, роботу з базою даних і інтеграцію із зовнішніми сервісами.

Для забезпечення гнучкості та масштабованості часто застосовується **архітектура RESTful API**, де сервер надає набір сервісів (API), а клієнтська частина звертається до них для отримання даних та виконання операцій. Це дозволяє розробляти мобільні додатки, вебзастосунки та інші клієнти на різних платформах, використовуючи спільний бекенд.

Крім того, сучасні вебзастосунки можуть використовувати **архітектуру мікросервісів**, що розбиває систему на незалежні сервіси, кожен із яких виконує певну функцію. Такий підхід підвищує надійність і спрощує масштабування, але вимагає складнішого адміністрування.

Важливо також врахувати використання **кешування**, що дозволяє зменшити навантаження на сервер і пришвидшити обробку запитів. Застосування технологій, таких як CDN (Content Delivery Network), покращує швидкість завантаження статичних ресурсів.

Обраний тип архітектури має підтримувати безперервне оновлення, простоту тестування і впровадження нових функцій, зокрема кастомізації рецептів і авторських смаків.

### 3.2. Технологічний стек для реалізації

Вибір технологічного стеку визначає інструменти, мови програмування та платформи, які будуть використані для розробки інтернет-магазину. Для

сучасного вебзастосування важливо обрати надійні, масштабовані та широко підтримувані технології.

Для фронтенду найчастіше застосовуються **JavaScript-фреймворки**: React, Vue.js або Angular. Вони дозволяють створювати динамічний, інтерактивний інтерфейс користувача з можливістю повторного використання компонентів. React популярний завдяки великій екосистемі і підтримці, що робить його оптимальним вибором для проєкту.

На бекенді доцільно використовувати **Node.js** із фреймворком Express, або популярні рішення на основі Python (Django, Flask), PHP (Laravel), або Java (Spring). Node.js добре підходить для реального часу і високої навантаженості, що актуально для онлайн-замовлень.

Для бази даних рекомендується обрати між **реляційними СУБД** (MySQL, PostgreSQL) для структурованих даних та **NoSQL** (MongoDB) для більш гнучкої роботи з даними. У випадку інтернет-магазину, де важлива цілісність даних і складні запити, перевагу часто віддають реляційним СУБД.

Для забезпечення безпеки застосовуються такі технології, як HTTPS, JWT (JSON Web Tokens) для аутентифікації, а також системи управління правами доступу.

Інструменти для розробки включають системи контролю версій (Git), CI/CD платформи для автоматизації тестування та розгортання, що підвищує якість коду і швидкість релізів.

### 3.3. Моделювання бази даних

Моделювання бази даних — це процес створення логічної структури даних, яка забезпечує ефективне зберігання, пошук та обробку інформації про піци, користувачів, замовлення і налаштування.

Основою моделі є таблиці (у реляційних базах даних) або колекції (у NoSQL). Важливо чітко визначити сутності та їхні зв'язки. Основні сутності для інтернет-магазину піци:

- **Користувачі** (з інформацією про логін, пароль, контактні дані, історію замовлень);

- **Піци** (стандартні рецепти, авторські, а також створені користувачами);
- **Інгредієнти** (назва, тип, харчова інформація, алергени);
- **Замовлення** (зв'язок з користувачем, статус, дата, сума);
- **Кошик** (тимчасове зберігання вибраних товарів);
- **Оплати** (спосіб, статус транзакції).

Між сутностями встановлюються зв'язки: наприклад, між піцою і інгредієнтами — багато до багатьох (адже одна піца містить багато інгредієнтів, а один інгредієнт може входити до багатьох піц).

Процес моделювання включає розробку ER-діаграм (Entity-Relationship), що наочно відображає структуру даних. Важливо передбачити нормалізацію бази даних, щоб уникнути дублювання інформації і забезпечити цілісність даних.

Особлива увага приділяється швидкості запитів, тому можливо застосування індексів і кешування.

### 3.4. Розробка структури інтерфейсу користувача

Інтерфейс користувача (UI) є ключовим елементом успіху інтернет-магазину, адже від його зручності та привабливості залежить рівень залученості і задоволеності клієнтів.

Структура інтерфейсу розробляється на основі аналізу потреб користувачів і функціональних вимог. Вона повинна забезпечувати простий і логічний шлях від входу до оформлення замовлення.

Основні компоненти UI:

- **Головна сторінка** — презентація бренду, акцій, швидкий доступ до меню.
- **Каталог піци** — з можливістю фільтрації (по типу, смаку, ціні), пошуку та перегляду деталей.
- **Конструктор піци** — інтуїтивний інтерфейс для вибору інгредієнтів, типу тіста, розміру.

- **Кошик і оформлення замовлення** — з чіткою індикацією вартості, варіантів доставки і оплати.
- **Особистий кабінет** — управління профілем, історія замовлень, збережені рецепти.
- **Сторінки допомоги та контакти** — для підтримки користувачів.

UI повинен бути адаптивним, щоб коректно відображатися на різних пристроях — ПК, планшетах, смартфонах. Дизайн має бути сучасним, лаконічним, із приємною кольоровою гамою, що підкреслює стиль бренду.

### 3.5. Прототипування конструктора піци

Прототипування конструктора піци — це створення первинної моделі інтерфейсу, яка дозволяє користувачам формувати власну піцу, вибираючи інгредієнти та налаштування.

Прототип допомагає протестувати ідеї, отримати відгуки користувачів і внести необхідні корективи на ранньому етапі розробки.

Основні функціональні елементи конструктора:

- Вибір типу тіста (тонке, товсте, безглютенове тощо);
- Вибір соусу (томатний, білий, песто тощо);
- Додавання сирів, м'яса, овочів, спецій;
- Можливість видалити інгредієнти або змінити їх кількість;
- Візуалізація піци в реальному часі;
- Збереження створеного рецепта для повторного замовлення.

Прототип може бути створений за допомогою спеціалізованих інструментів (Figma, Adobe XD, Sketch), що дозволяють розробляти інтерактивні макети.

У процесі тестування важливо звернути увагу на простоту використання, швидкість відгуку і логіку розміщення елементів.

### 3.6. Безпека та захист персональних даних

Захист персональних даних користувачів і безпека інтернет-магазину — критично важливі аспекти, які повинні бути враховані у процесі розробки.

Вебзастосунок повинен відповідати вимогам законодавства щодо захисту даних (наприклад, GDPR для європейських користувачів) і включати механізми:

- **Шифрування даних** — використання HTTPS для захищеного зв'язку між клієнтом і сервером, а також збереження паролів у базі даних у хешованому вигляді (bcrypt, Argon2);
- **Аутентифікація і авторизація** — використання надійних методів, таких як JWT або OAuth, з обмеженням доступу до особистих даних;
- **Захист від атак** — впровадження механізмів проти SQL-ін'єкцій, XSS (міжсайтового скриптингу), CSRF (підробки міжсайтових запитів);
- **Регулярні оновлення і патчі** — підтримка системи в актуальному стані для усунення вразливостей;
- **Політика конфіденційності** — прозора інформація для користувачів про збір, зберігання та використання їхніх даних.

Також важливо навчити персонал правилам роботи з конфіденційною інформацією та створити систему резервного копіювання даних.

## РОЗДІЛ 4. РЕАЛІЗАЦІЯ ІНТЕРНЕТ-МАГАЗИНУ ПІЦЦИ

### 4.1. Створення клієнтської частини (Frontend)

Клієнтська частина вебзастосунку відповідає за взаємодію користувача з системою, відображення інформації та збір даних. Для інтернет-магазину піци було обрано сучасний JavaScript-фреймворк React, що забезпечує динамічність, реактивність і модульність інтерфейсу.

Процес розробки фронтенду розпочинався зі створення каркасної структури компонентів, що відповідають за різні частини інтерфейсу: головна сторінка, каталог піци, конструктор, кошик, профіль користувача та оформлення замовлення. Для управління станом застосунку використовувався Redux, що дозволяє централізовано зберігати та оновлювати інформацію, наприклад, вміст кошика або вибрані інгредієнти.

Особлива увага була приділена адаптивному дизайну, щоб інтерфейс коректно відображався на різних пристроях: десктопах, планшетах і смартфонах. Для стилізації використано CSS-модулі та бібліотеки, такі як Material-UI, що спрощують розробку привабливого і зручного UI.

Впроваджено систему валідації форм, зручний пошук і фільтри для каталогу. Для покращення UX реалізовані анімації завантаження та повідомлення про помилки чи успішні дії.

Таким чином, клієнтська частина стала зручною, швидкою і інтуїтивно зрозумілою для користувачів.

### 4.2. Розробка серверної частини (Backend)

Серверна частина забезпечує логіку обробки замовлень, управління даними та взаємодію з базою даних. Для бекенду було обрано Node.js з Express — це дозволяє ефективно обробляти одночасні запити і підтримувати масштабованість системи.

Бекенд реалізує RESTful API, через яке фронтенд отримує необхідні дані і відправляє команди. Впроваджено модулі для керування користувачами (реєстрація, логін, автентифікація), обробки меню (отримання списку піц, додавання авторських рецептів), а також управління замовленнями та оплатою.

Застосовано JWT для безпечної автентифікації користувачів і захисту особистої інформації. Для роботи з базою даних використано ORM-бібліотеку Sequelize, що спрощує взаємодію із реляційною СУБД PostgreSQL.

Сервер обробляє запити на створення, оновлення та видалення замовлень, надсилає статуси користувачам і керує акціями та знижками.

Розробка бекенду включала також реалізацію логування, обробку помилок та забезпечення безперервної роботи через налаштування середовища розгортання.

### **4.3. Реалізація конструктора піци**

Конструктор піци — одна з ключових функцій системи, що дозволяє користувачам створювати індивідуальні рецепти. Реалізація цього компонента включала створення інтуїтивного інтерфейсу, де користувач вибирає тип тіста, соус, сири, м'ясо, овочі та спеції.

У фронтенді було розроблено інтерактивну панель вибору інгредієнтів із відображенням їхніх характеристик (калорійність, алергени) і візуалізацією піци в режимі реального часу. Користувач може додавати, видаляти або змінювати кількість інгредієнтів.

Для бекенду створено логіку перевірки валідності вибору (наприклад, обмеження на кількість інгредієнтів), обчислення вартості та формування замовлення.

Реалізовано можливість збереження створених піц у профілі користувача для подальшого швидкого замовлення.

Тестування конструктора показало зручність та простоту використання, що позитивно впливає на користувацький досвід.

### **4.4. Додавання та управління авторськими рецептами**

Авторські рецепти відображають унікальність бренду, тому розробка функції додавання та управління ними є важливим етапом.

У адміністративній панелі реалізовано інтерфейс для створення нових рецептів, редагування та видалення існуючих. Адміністратори можуть додавати

опис, фотографії, склад інгредієнтів, ціни, а також позначати рецепти як сезонні або акційні.

Для користувачів авторські рецепти доступні у каталозі, із можливістю швидкого додавання до кошика або кастомізації.

Бекенд забезпечує збереження даних у базі, контролює права доступу, валідацію інформації та пошук за різними параметрами.

Забезпечена можливість аналітики популярності авторських рецептів для подальшого поліпшення асортименту.

#### **4.5. Система обробки замовлень**

Система обробки замовлень є центральним модулем інтернет-магазину, що забезпечує прийом, збереження, обробку та контроль виконання замовлень.

При оформленні замовлення користувач вводить адресу доставки, контактні дані, обирає спосіб оплати. Система перевіряє коректність даних і створює запис у базі.

Далі замовлення потрапляє до черги обробки для кухні і служби доставки. Реалізовано автоматичне оновлення статусів (прийняте, готується, в дорозі, доставлено), що відображається у особистому кабінеті користувача.

Адміністратор має інтерфейс для перегляду та управління замовленнями, зміни статусів і роботи зі звітністю.

Також передбачено обробку відмов, повернень і зворотного зв'язку від клієнтів.

#### **4.6. Тестування основних функцій системи**

Тестування — обов'язковий етап, який гарантує якість і надійність системи. В ході розробки проводилось кілька видів тестування:

- **Функціональне тестування**, яке перевіряло відповідність реалізованих функцій заявленим вимогам (реєстрація, оформлення замовлень, робота конструктора);
- **Юзабіліті-тестування**, що дозволяло оцінити зручність інтерфейсу, логіку взаємодії та швидкість реакції системи;

- **Тестування безпеки**, спрямоване на виявлення вразливостей, захист від SQL-ін'єкцій, XSS і CSRF атак;
- **Тестування продуктивності** — оцінка швидкості роботи при великій кількості одночасних користувачів.

Для автоматизації тестів було використано Jest для фронтенду і Mocha/Chai для бекенду.

Результати тестування показали стабільність системи і високу якість реалізації основних функцій, що забезпечує позитивний користувацький досвід.

## ВИСНОВКИ

У ході виконання даної дипломної роботи було розглянуто та реалізовано інтернет-магазин піци з унікальним функціоналом, що передбачає можливість створення індивідуальних рецептів та використання авторських смаків. Реалізація цього проєкту відповідає сучасним тенденціям ринку онлайн-замовлень їжі, що активно розвивається, і враховує потреби різноманітної цільової аудиторії.

Перш за все, було проаналізовано сучасний стан ринку онлайн-замовлень піци. Виявлено, що на сьогоднішній день користувачі очікують не тільки швидкість і зручність замовлення, а й індивідуалізацію продукту, можливість експериментувати зі смаками та отримувати якісний сервіс. Конкурентний аналіз показав, що більшість існуючих платформ пропонує стандартний набір піц, і лише деякі з них мають обмежений функціонал кастомізації. Це створює сприятливі умови для реалізації проєкту з гнучкими авторськими рецептами та конструктором піци.

В результаті дослідження потреб цільової аудиторії було визначено основні вимоги до системи: простота у використанні, широкий вибір інгредієнтів, наявність докладної інформації про склад і харчову цінність, а також надійна система обробки замовлень. Унікальність у поєднанні з інтуїтивно зрозумілим інтерфейсом забезпечує високий рівень лояльності користувачів і конкурентні переваги.

У процесі розробки було обрано сучасний технологічний стек, що включає React для клієнтської частини, Node.js з Express — для серверної, та PostgreSQL — для зберігання даних. Використання цих технологій дозволило створити масштабовану, швидку та безпечну систему. Архітектура застосунку побудована за принципом клієнт-сервер, що полегшує подальше обслуговування та розширення функціоналу.

Особливу увагу приділено розробці конструктора піци, який дозволяє користувачам формувати унікальні рецепти, вибираючи з широкого набору інгредієнтів, а також зберігати їх для подальшого замовлення. Цей функціонал

підтримує тенденції персоналізації продуктів і сприяє залученню аудиторії, яка цінує індивідуальний підхід.

Для забезпечення безпеки та конфіденційності користувачів реалізовано надійну систему автентифікації та авторизації на основі JWT, застосовано захист від основних веб-атак (SQL-ін'єкції, XSS, CSRF), а також впроваджено шифрування паролів. Це гарантує захист персональних даних та довіру користувачів до платформи.

Розроблена система обробки замовлень автоматизує процес від прийому заявки до доставки, з можливістю відстеження статусу в режимі реального часу. Завдяки цьому підвищується прозорість і контроль якості обслуговування, що позитивно впливає на задоволеність клієнтів.

Тестування системи показало стабільність і надійність реалізованих модулів. Функціональні тести підтвердили відповідність системи вимогам, юзабіліті-тестування засвідчило зручність користувацького інтерфейсу, а тести безпеки підтвердили захищеність від поширених загроз. Продуктивність платформи дозволяє ефективно обслуговувати значну кількість одночасних користувачів.

Практичне значення отриманих результатів полягає у створенні готового до впровадження інструменту для бізнесу, який може суттєво розширити можливості традиційних піцерій та підвищити їх конкурентоспроможність на ринку. Запропонований підхід до кастомізації продуктів може бути адаптований і для інших напрямів онлайн-торгівлі.

В подальших дослідженнях рекомендується розглянути впровадження системи рекомендацій на основі машинного навчання, що аналізуватиме смаки користувачів та пропонуватиме індивідуальні поєднання інгредієнтів. Також можливим є розширення платформи за рахунок інтеграції з мобільними додатками та популярними платіжними сервісами.

Таким чином, виконана робота демонструє комплексний підхід до створення сучасного інтернет-магазину піци, який враховує як бізнес-потреби, так і побажання кінцевих користувачів. Впровадження розробленої системи

дозволить підвищити якість обслуговування, збільшити обсяг продажів і зміцнити позиції компанії на ринку.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бобровський І.В. Веб-програмування : навч. посіб. / І.В. Бобровський. — К. : Логос, 2019. — 320 с.
2. Петрова О.В. Розробка веб-застосунків на React.js : підручник / О.В. Петрова. — Львів : Астролябія, 2021. — 280 с.
3. Іваненко С.М. Архітектура програмного забезпечення / С.М. Іваненко. — Харків : Фоліо, 2018. — 240 с.
4. Курило В.І. Безпека веб-застосунків / В.І. Курило. — К. : Центр учбової літератури, 2020. — 196 с.
5. Гнатюк Д.В. Проектування баз даних / Д.В. Гнатюк. — Львів : Новий Світ, 2017. — 212 с.
6. Smith J., Brown M. Building Secure Web Applications. — New York : TechPress, 2022. — 350 p.
7. Mozilla Developer Network. Web Security Guidelines [Електронний ресурс]. — Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/Security> (дата звернення: 15.05.2025).
8. PostgreSQL Documentation [Електронний ресурс]. — Режим доступу: <https://www.postgresql.org/docs/> (дата звернення: 16.05.2025).
9. React Official Documentation [Електронний ресурс]. — Режим доступу: <https://reactjs.org/docs/getting-started.html> (дата звернення: 14.05.2025).
10. Node.js Documentation [Електронний ресурс]. — Режим доступу: <https://nodejs.org/en/docs/> (дата звернення: 14.05.2025).
11. Express.js Guide [Електронний ресурс]. — Режим доступу: <https://expressjs.com/en/guide/routing.html> (дата звернення: 14.05.2025).
12. Auth0. Authentication and Authorization Concepts [Електронний ресурс]. — Режим доступу: <https://auth0.com/docs> (дата звернення: 17.05.2025).
13. Nielsen J. Usability Engineering. — San Francisco : Morgan Kaufmann, 1994. — 368 p.

14. Gartner Report. Online Food Delivery Trends 2024 [Електронний ресурс]. — Режим доступу: <https://gartner.com/food-delivery-report-2024> (дата звернення: 12.05.2025).

15. Коваленко Т.В. Маркетингові дослідження в інтернет-бізнесі / Т.В. Коваленко. — К. : Майбутнє, 2020. — 190 с.

16. ChatGPT (OpenAI). Консультації з розробки інтернет-магазину піци з кастомізацією рецептів [Електронний ресурс]. — Режим доступу: <https://openai.com/chatgpt> (дата звернення: 22.05.2025).

17. Shneiderman B. Designing the User Interface: Strategies for Effective Human-Computer Interaction. — Pearson, 2016. — 560 p.

## ДОДАТКИ

### Додаток А. Код клієнтської частини (Frontend)

#### 1. Компоненти відображення товарів

```
export default function RecipeListItem({ recipe }) {  
  
  return (  
  
    <div>  
  
      <div className='sm:flex w-full p-5 h-fit rounded-xl border-purple-300 border-2 my-5 hidden  
        bg-[#221B24]'  
  
        <div className='relative w-[20%] h-fit'  
  
          <img  
  
            src={recipe.photo_url}  
  
            alt="."  
  
            className="object-cover rounded-xl transition-all duration-300"  
  
          />  
  
        </div>  
  
        <div className=" text-white w-[80%]">  
  
          <h4 className="text-xl font-bold text-center px-2">  
  
            {recipe.name}  
  
          </h4>  
  
          <div className='flex m-5'  
  
            <p>  
  
              <span className='text-[#8742A0]';>Опис :</span> {recipe.description}  
  
            </p>  
  
          </div>  
  
          <div className='flex justify-between'  
  
            <p className="p-5 text-3xl font-bold text-[#AEF414] flex">  
  
              {recipe.price} грн
```

```

    </p>
    <a href={` /recipe/public/${recipe.slug}`} className="my-auto p-2 rounded-xl bg-[#8742A0]">
      Детальніше
    </a>
  </div>
</div>
<div className="block w-full h-fit text-white rounded-xl border-purple-300 border-2
my-5 sm:hidden bg-[#221B24]">
  <h4 className="text-xl font-bold text-center p-2">
    {recipe.name}
  </h4>
  <div className="relative mx-5 h-fit">
    <img
      src={recipe.photo_url}
      alt="."
      className="object-cover rounded-xl transition-all duration-300"
    />
  </div>
  <div className="m-5">
    <span className="text-[#8742A0]">Опис :</span> {recipe.description}
  </div>
  <div className="flex">
    <p className="text-xl font-bold text-[#AEF414] ml-auto p-5">
      {recipe.price} грн
    </p>
  </div>
  <a className="p-4 bg-[#8742A0] w-full flex rounded-b-xl" href={` /recipe/public/${recipe.slug}`}>

```

```

    <p className="mx-auto">Детальніше</p>
  </a>
</div>
</div>
);
}

```

## 2. Компоненти замовлення

```

import { Link } from "react-router-dom";
import { useState } from "react";

export default function OrderItem({ order, onDeleteClick }) {
  const [isImageHovered, setIsImageHovered] = useState(false);

  return (
    <div className="bg-[rgba(135,66,160,0.25)] rounded-xl transition-all duration-300 my-2 sm:my-5">
      <div className="sm:flex hidden">
        <div className="ml-[5%] sm:w-[20%] w-[50%] p-2">
          <div className="flex">
            <Link
              to={` /recipe/public/${order.recipe_slug} `}
              className="relative sm:w-50 w-25"
              onMouseEnter={() => setIsImageHovered(true)}
              onMouseLeave={() => setIsImageHovered(false)}
            >
              <img
                src={order.recipe_photo_url}
                alt={order.recipe_name}
                className={`w-full sm:h-35 h-20 object-cover mt-[10%] rounded-xl transition-all duration-300 $`

```

```

        isImageHovered ? "brightness-50" : ""
      }}
    />

    {isImageHovered && (
      <div className="absolute inset-0 flex items-center justify-center">
        <span className="text-[#AEF414] sm:text-xl text-sm font-bold bg-black bg-opacity-50 px-4 mt-[10%] py-
2 rounded-xl">
          Перейти
        </span>
      </div>
    )}
  </Link>
</div>
</div>

<div className="w-[55%] pl-[1%] pr-[5%] text-white flex">
  <div className="m-auto pt-3 text-2xl">
    <p className="m-auto pt-3 text-2xl">{order.recipe_name}</p>
    <p className="text-[#AEF414] text-center font-bold">
      {order.recipe_price > 0 ? `${order.recipe_price} грн` : "Безкоштовно"}
    </p>
  </div>
</div>

<div className="w-[30%] text-white bg-[#8742A0] rounded-xl flex items-center justify-center">
  <button
    onClick={() => onDeleteClick(order.id)}
    className="bg-[#AEF414] text-black font-medium rounded-xl p-4 h-fit w-fit text-center hover:bg-[#8bc11a]
transition-colors"
  >

```

```

>
  <p>Видалити</p>
</button>
</div>
</div>

<div className="sm:hidden text-white">

  <div className="flex p-2">

    <div className="flex">

      <Link

        to={`/recipe/public/${order.recipe_slug}`}

        className="relative w-25"

        onMouseEnter={() => setIsImageHovered(true)}

        onMouseLeave={() => setIsImageHovered(false)}

      >

        <img

          src={order.recipe_photo_url}

          alt={order.recipe_name}

          className={`mb-2 w-fit h-25 object-cover rounded-xl transition-all duration-300 ${

            isImageHovered ? "brightness-50" : ""

          }}

        />

        {isImageHovered && (

          <div className="absolute inset-0 flex items-center justify-center">

            <span className="text-[#AEF414] text-sm font-bold bg-black bg-opacity-50 px-4 mt-[10%] py-2 rounded-

xl">

              Перейти

            </span>

          </div>

        )}

      </div>

```

```

    })
  </Link>

  <div className="p-4">

    <p className="text-xl">{order.recipe_name}</p>

    <p className="text-[#AEF414] font-bold">
      {order.recipe_price > 0 ? `${order.recipe_price} грн` : "Безкоштовно"}
    </p>

  </div>

</div>

</div>

<div className="w-full text-white bg-[#8742A0] rounded-xl flex my-3">

  <button

    onClick={() => onDeleteClick(order.id)}

    className="bg-[#AEF414] text-black font-medium rounded-xl p-3 h-fit w-fit mx-auto my-3 hover:bg-
[#8bc11a] transition-colors"

    >

    <p>Видалити</p>

  </button>

</div>

</div>

</div>
);
}

```

### 3. Сторінкова навігація (пагінація)

```

import { useState } from 'react';

import pagPrev from '../files_photo/pag_prev.png';

import pagPrevHover from '../files_photo/pag_prev_hover.png';

import pagNext from '../files_photo/pag_next.png';

```

```
import pagNextHover from '../files_photo/pag_next_hover.png';

export default function PaginationLinks({ meta, onPageClick }) {

  const [prevImg, setPrevImg] = useState(pagPrev);

  const [nextImg, setNextImg] = useState(pagNext);

  if (!meta) return null;

  function onClick(ev, link) {

    ev.preventDefault();

    if (!link.url) return;

    onPageClick(link);

  }

  return (

    <div style={{ backgroundColor: '#302633' }} className="flex items-center justify-between px-4 py-3 sm:px-6 mt-4">

      <div className="flex flex-1 justify-between sm:hidden">

        <button onClick={ev => onClick(ev, meta.links[0])} className="relative inline-flex items-center">

          <img

            src={prevImg}

            onMouseOver={() => setPrevImg(pagPrevHover)}

            onMouseOut={() => setPrevImg(pagPrev)}

            alt="Previous"

            className="w-10 h-10"

          />

        </button>


```

```

<button onClick={ev => onClick(ev, meta.links[meta.links.length - 1])} className="relative ml-3 inline-flex
items-center">

  <img

    src={nextImg}

    onMouseOver={() => setNextImg(pagNextHover)}

    onMouseOut={() => setNextImg(pagNext)}

    alt="Next"

    className="w-10 h-10"

  />

</button>

</div>

<div className="hidden sm:flex sm:flex-1 sm:items-center sm:justify-center">

  <div>

    {meta.total > meta.per_page && (

      <nav className="isolate inline-flex -space-x-px" aria-label="Pagination">

        {/* Кнопка "Previous" */}

        <button onClick={ev => onClick(ev, meta.links[0])} className="mr-2">

          <img

            src={prevImg}

            onMouseOver={() => setPrevImg(pagPrevHover)}

            onMouseOut={() => setPrevImg(pagPrev)}

            alt="Previous"

            className="w-10 h-10"

          />

        </button>

        {meta.links.map((link, ind) => {

```

```
if (link.label.includes('Previous') || link.label.includes('Next')) return null;
```

```
return (
```

```
<a
```

```
  href="#"
```

```
  onClick={ev => onClick(ev, link)}
```

```
  key={ind}
```

```
  aria-current={link.active ? "page" : undefined}
```

```
  className={
```

```
    `relative z-10 inline-flex items-center px-4 py-2 font-medium focus:z-20
```

```
    transition-colors duration-200 text-2xl
```

```
    ${ind === 1 ? 'rounded-l-md' : "}"
```

```
    ${ind === meta.links.length - 2 ? 'rounded-r-md' : "}"
```

```
    ${link.active
```

```
      ? 'text-lime-400'
```

```
      : 'text-purple-600 hover:text-lime-400'}`
```

```
  }
```

```
  dangerouslySetInnerHTML={{ __html: link.label }}
```

```
>
```

```
);
```

```
}}}
```

```
<button onClick={ev => onClick(ev, meta.links[meta.links.length - 1])} className="ml-2">
```

```
<img
```

```
  src={nextImg}
```

```
  onMouseOver={() => setNextImg(pagNextHover)}
```

```
  onMouseOut={() => setNextImg(pagNext)}
```

```
  alt="Next"
```

```

        className="w-10 h-10"

      />

    </button>

  </nav>

  })

</div>

</div>

</div>

);
}

```

#### 4. Компоненти керування сторінками

```

import { Disclosure, DisclosureButton, DisclosurePanel, Menu, MenuButton, MenuItem,
MenuItemItems } from '@headlessui/react'

import { Link, Navigate, NavLink, Outlet } from 'react-router-dom'

import { useStateContext } from '../contexts/ContextProvider'

import axiosClient from '../axios'

import logo from '../files_photo/logo.png';

import menu from "../files_photo/icon_menu.png"

import cross from "../files_photo/cross.png"

import { useState } from "react";

import { useEffect } from "react";

const navigation = [

  { name: 'Рецепти', to: '/'},

  { name: 'Про проект', to: '/about'}

]

```

```
function classNames(...classes) {
  return classes.filter(Boolean).join(' ')
}

export default function DefaultLayout() {
  const {currentUser, userToken, setCurrentUser, setUserToken} = useStateContext();
  const [isOpen, setIsOpen] = useState(false);

  useEffect(() => {
    axiosClient.get('/me')
      .then(({ data }) => {
        setCurrentUser(data)
      })
  }, [])

  if (!userToken){
    return <Navigate to='/login'/>
  }

  const logout = (ev) => {
    ev.preventDefault();
    axiosClient.post('/logout')
      .then(res => {
        setCurrentUser({});
        setUserToken(null);
      })
  }
}
```

```

    });
  }

  return (
    <◇>
    <div className="min-h-full">
      <Disclosure as="nav" style={{ backgroundColor: '#221B24' }}>
        <div className="mx-auto max-w-7xl px-4 sm:px-6 lg:px-8">
          <div className="flex h-16 items-center justify-between">
            <div className="flex items-center">
              <div className="shrink-0">
                <Link to="/">
                  <img
                    alt="."
                    src={logo}
                    className="w-10"
                  />
                </Link>
              </div>
              <div className="hidden md:block">
                <div className="ml-10 flex items-baseline space-x-4">
                  {navigation.map((item) => (
                    <NavLink
                      key={item.name}
                      to={item.to}
                      aria-current={item.current ? 'page' : undefined}

```

400'

```

className={({isActive}) => classNames(
  isActive ? 'shadow-[-2px_2px_0px_#8742A0]' : 'hover:text-[#AEF414]',
  'rounded-md px-3 py-2 text-sm font-medium text-[#8742A0] transition-all duration-
))
>
  {item.name}
</NavLink>
))}
</div>
</div>
</div>
<div className="hidden md:block">
  <div className="ml-4 flex items-center md:ml-6">
    <Menu as="div" className="relative ml-3">
      <div className='flex'>
        <div className='flex'>
          {currentUser.name && (
            <Link className='flex p-2' onClick={() => setIsOpen(true)}>
              <p className='text-white'>{currentUser.name}</p>
            </Link>
          )}
          {!currentUser.name && (
            <a className='flex p-2' href="#" onClick={(ev) => logout(ev)}>
              <p className='text-white'>Увійти</p>
            </a>
          )}
        </div>
      </div>
    </Menu>
  </div>
</div>

```

```

        </a>
    )}
</div>

    {isOpen && (
        <div className="absolute right-0 w-60 bg-[#452954] text-[#8742A0] rounded-lg
shadow-lg px-8 z-50 mt-13 py-5">
            <div className="flex justify-end ">
                <button onClick={() => setIsOpen(false)}>
                    <img src={cross} alt="Close" className="w-6 h-6" />
                </button>
            </div>
            <ul className="space-y-3 font-medium">
                <li>
                    <Link className="hover:text-[#AEF414]"
to={`/profile/${currentUser.id}`} `>Обліковий запис</Link>
                </li>
                <li>
                    <Link className="hover:text-[#AEF414]"
to={`/orders/${currentUser.id}`} `>Мої замовлення</Link>
                </li>
                <li>
                    <a className="hover:text-[#AEF414]" href="#" onClick={(ev) =>
logout(ev)}>Вихід</a>
                </li>
            </ul>
        </div>
    )}

```

```

</div>

<MenuItems
  transition
  className="absolute right-0 z-10 mt-2 w-48 origin-top-right rounded-md
  bg-white py-1 ring-1 shadow-lg ring-black/5 transition focus:outline-hidden
  data-closed:scale-95 data-closed:transform data-closed:opacity-0
  data-enter:duration-100 data-enter:ease-out data-leave:duration-75 data-leave:ease-
in"
  >

  <MenuItem>
    <a
      href="#"
      onClick={(ev) => logout(ev)}
      className="block px-4 py-2 text-sm text-gray-700 data-focus:bg-gray-100 data-
focus:outline-hidden"
    >
      Вийти
    </a>
  </MenuItem>
</MenuItems>
</Menu>
</div>

</div>
<div className="-mr-2 flex md:hidden">
  <DisclosureButton className="group relative">
    <span className="absolute -inset-0.5" />
    <span className="sr-only">Меню</span>

```

```

    <img src={menu} alt="." style={{width: "50%", margin: "auto"}}/>

  </DisclosureButton>

</div>

</div>

</div>

<DisclosurePanel className="md:hidden">

  <div className="space-y-1 px-2 pt-2 pb-3 sm:px-3">

    <Link className='flex p-2'>

      <div className='flex'>

        {currentUser.name && (

          <Link className='flex p-2' onClick={() => setIsOpen(true)}>

            <p className='text-white'>{currentUser.name}</p>

          </Link>

        )}

        {!currentUser.name && (

          <a className='flex p-2' href="#" onClick={(ev) => logout(ev)}>

            <p className='text-white'>Увійти</p>

          </a>

        )}

      </div>

    </Link>

    {isOpen && (

      <div className="absolute right-0 w-full bg-[#452954] text-[#8742A0] rounded-xl
shadow-lg px-8 z-50 py-5">

        <div className="flex justify-end ">

```

```

    <button onClick={() => setIsOpen(false)}>
      <img src={cross} alt="Close" className="w-6 h-6" />
    </button>
  </div>

  <ul className="space-y-3 font-medium">
    <li>
      <Link className="hover:text-[#AEF414]"
to={`/profile/${currentUser.id}`}>Обліковий запис</Link>
    </li>
    <li>
      <Link className="hover:text-[#AEF414]" to={`/orders/${currentUser.id}`}>Мої
замовлення</Link>
    </li>
    <li>
      <a className="hover:text-[#AEF414]" href="#" onClick={(ev) =>
logout(ev)}>Вихід</a>
    </li>
  </ul>
</div>
)}

{navigation.map((item) => (
  <NavLink
    key={item.name}
    to={item.to}
    className={({isActive}) => classNames(
      isActive ? 'shadow-[-2px_2px_0px_#8742A0]' : 'hover:text-[#AEF414]',
      'mt-3 ml-1 w-fit block rounded-xl px-3 py-2 text-base font-medium text-[#8742A0]
transition-all duration-400',

```

```

        })
    >
        {item.name}
    </NavLink>
    ))}
</div>
</DisclosurePanel>
</Disclosure>
<Outlet />
</div>
</>
)
}

```

## Додаток Б. Код серверної частини (Backend)

### 1. Роутинг для товарів (піци)

```

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\RecipeController;

Route::middleware('auth:sanctum')->group(function () {
    Route::apiResource('recipe', RecipeController::class);
});

Route::get('/recipe/get-by-slug/{recipe:slug}', [RecipeController::class, 'getBySlug']);
Route::post('/recipe/increase-views/{id}', [RecipeController::class, 'increaseViews']);

```

### 2. Роутинг для замовлень

```

use App\Http\Controllers\OrderController;

use App\Http\Controllers\UserController;

use App\Http\Controllers\AuthController;

Route::middleware('auth:sanctum')->group(function () {

    // ЗАМОВЛЕННЯ

    Route::apiResource('order', OrderController::class);

    Route::get('/order/{id}', [OrderController::class, 'show']);

    Route::get('/orders-for-executor', [OrderController::class, 'ordersForExecutor']);

    // Користувачі

    Route::get('/profile/{id}', [UserController::class, 'show']);

    Route::put('/profile/{id}', [UserController::class, 'updateProfile']);

    Route::get('/me', [AuthController::class, 'me']);

    // Вихід

    Route::post('/logout', [AuthController::class, 'logout']);

});

// Публічні маршрути авторизації

Route::post('/signup', [AuthController::class, 'signup']);

Route::post('/login', [AuthController::class, 'login']);

```

### 3. Приклад структури даних

#### 3.1. Модель Recipe (піца)

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

```

```

use Illuminate\Support\Str;

class Recipe extends Model
{
    use HasFactory;

    protected $fillable = ['name', 'description', 'photo', 'user_id', 'created_at', 'updated_at', 'slug', 'price', 'views'];

    protected static function boot()
    {
        parent::boot();

        static::creating(function ($recipe) {
            $recipe->slug = static::generateUniqueSlug($recipe->name);
        });
    }

    protected static function generateUniqueSlug($name)
    {
        $slug = Str::slug($name);
        $count = static::where('slug', 'LIKE', "$slug%")->count();

        return $count ? "{$slug}-{$count}" : $slug;
    }

    public function user()
    {
        return $this->belongsTo(User::class);
    }
}

```

### 3.2. Модель Order (заказа)

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Order extends Model
{
    use HasFactory;
    protected $table = 'orders';
    protected $fillable = [
        'user_id',
        'id_of_recipe'
    ];

    public function user()

```

```

    {
        return $this->belongsTo(User::class);
    }

    public function recipe()
    {
        return $this->belongsTo(Recipe::class, 'id_of_recipe');
    }
}

```

### 3.3. Модель User (користувач)

```

<?php

namespace App\Models;

use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'email',
        'password',
        'phone'
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * The attributes that should be cast.

```

```

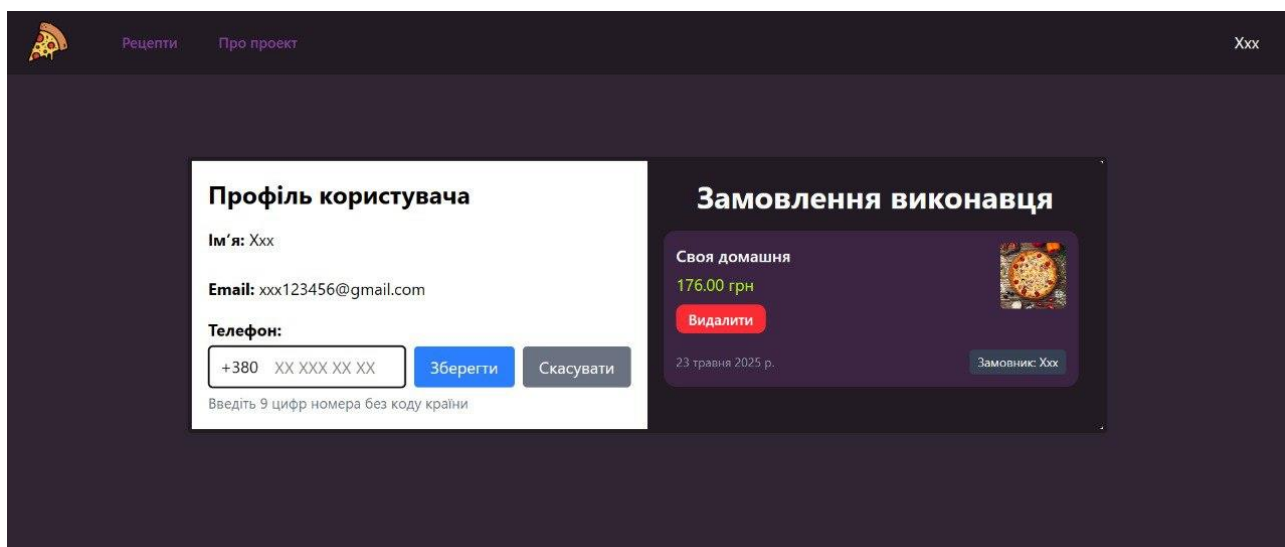
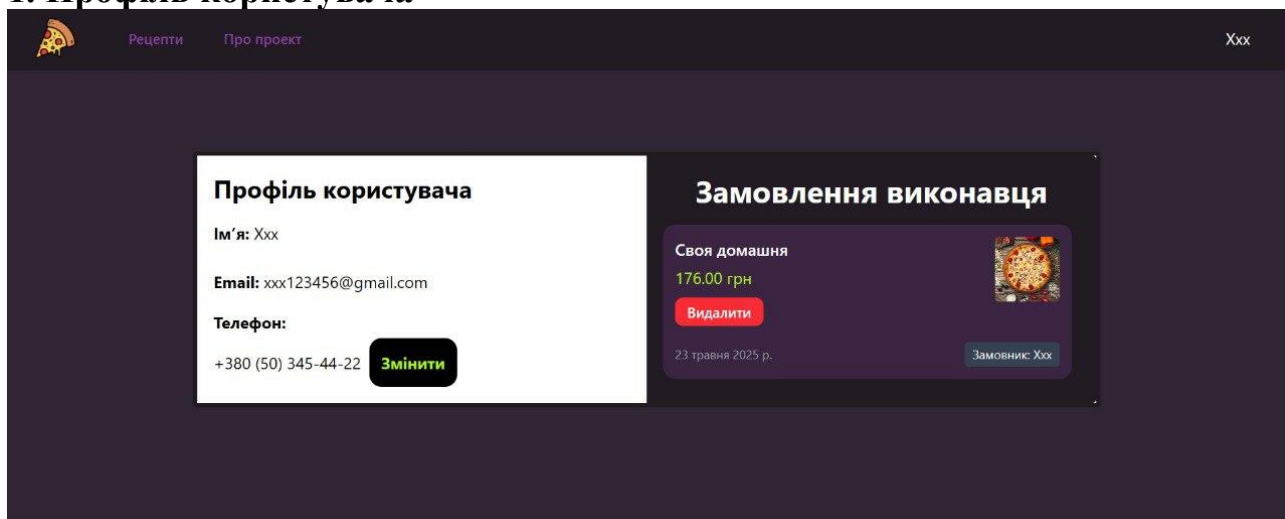
*
* @var array<string, string>
*/
protected $casts = [
    'email_verified_at' => 'datetime',
];

public function wishlist()
{
    return $this->belongsToMany(Game::class, 'wishlists');
}
}

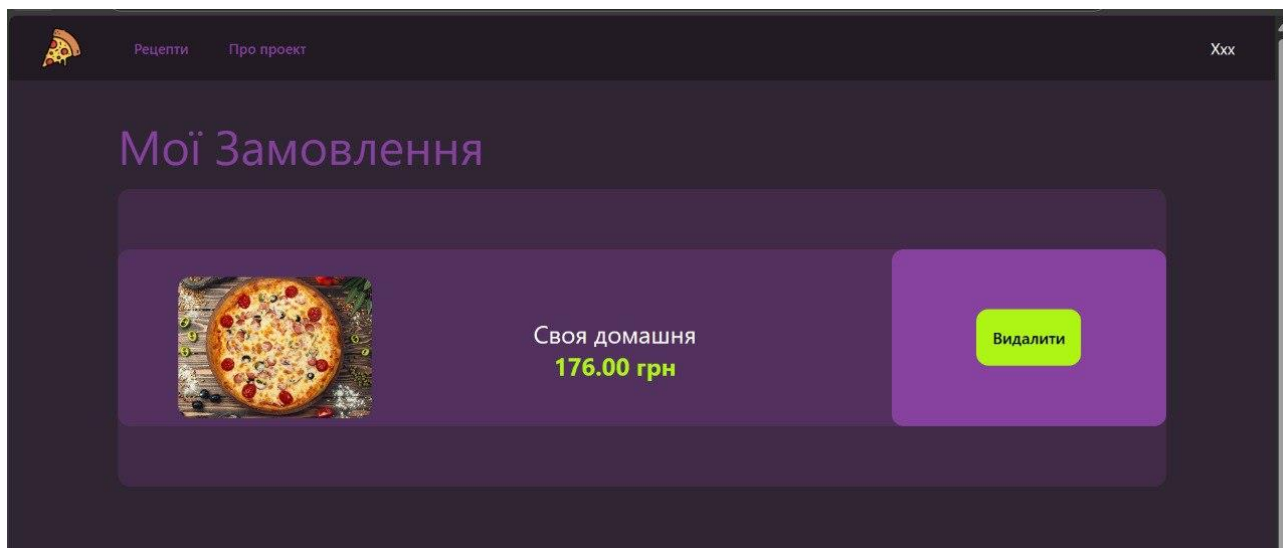
```

## Додаток В. Скріншоти розробленого сайту

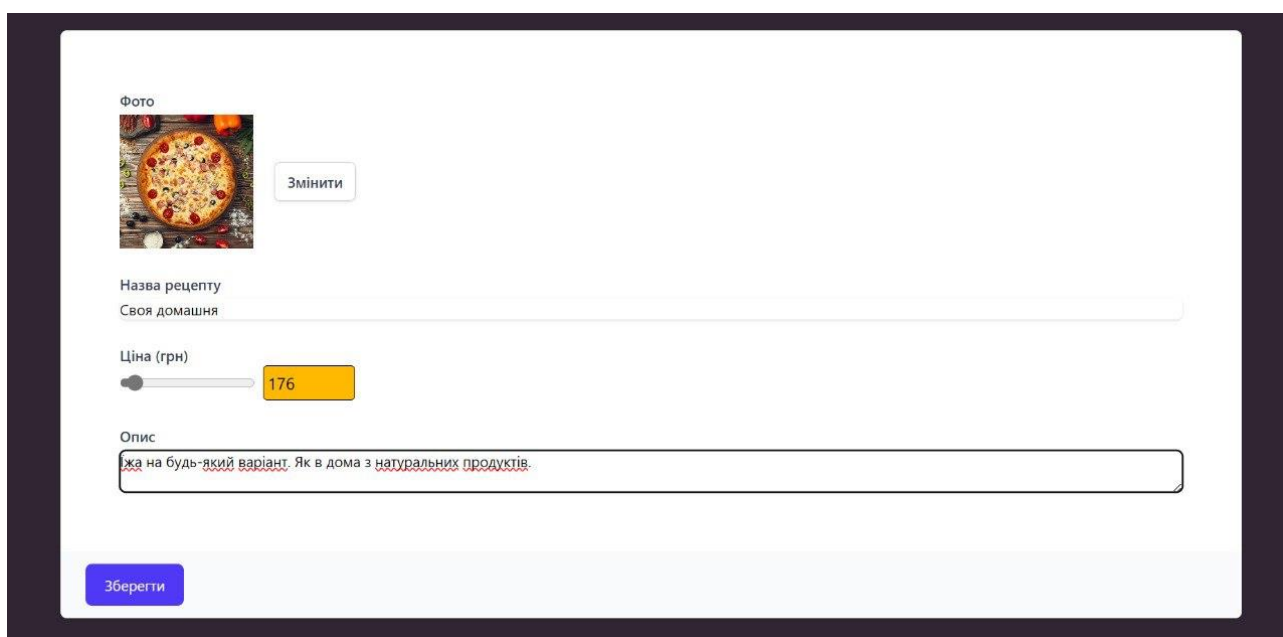
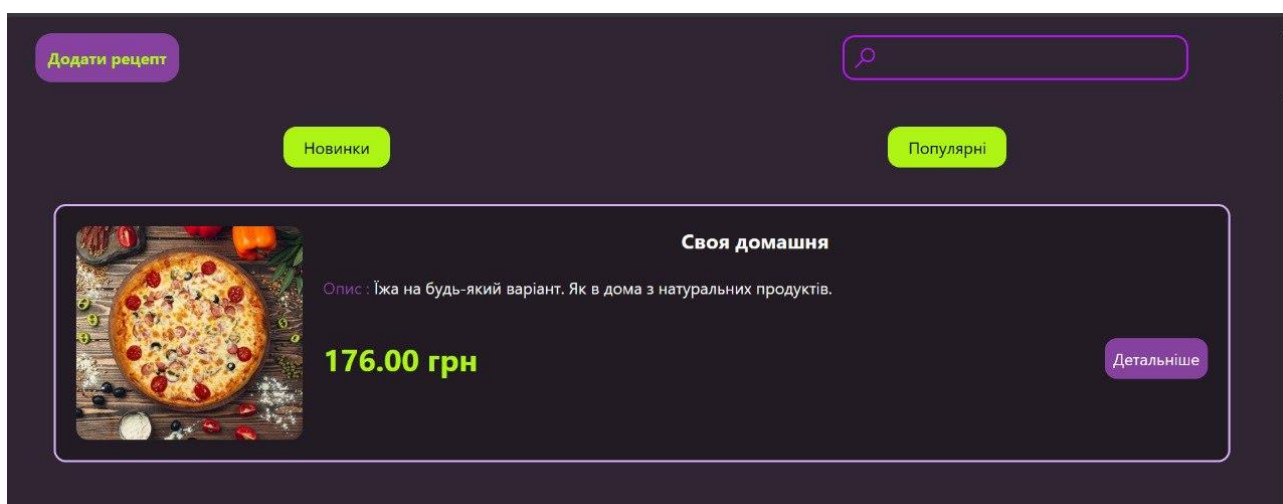
### 1. Профіль користувача



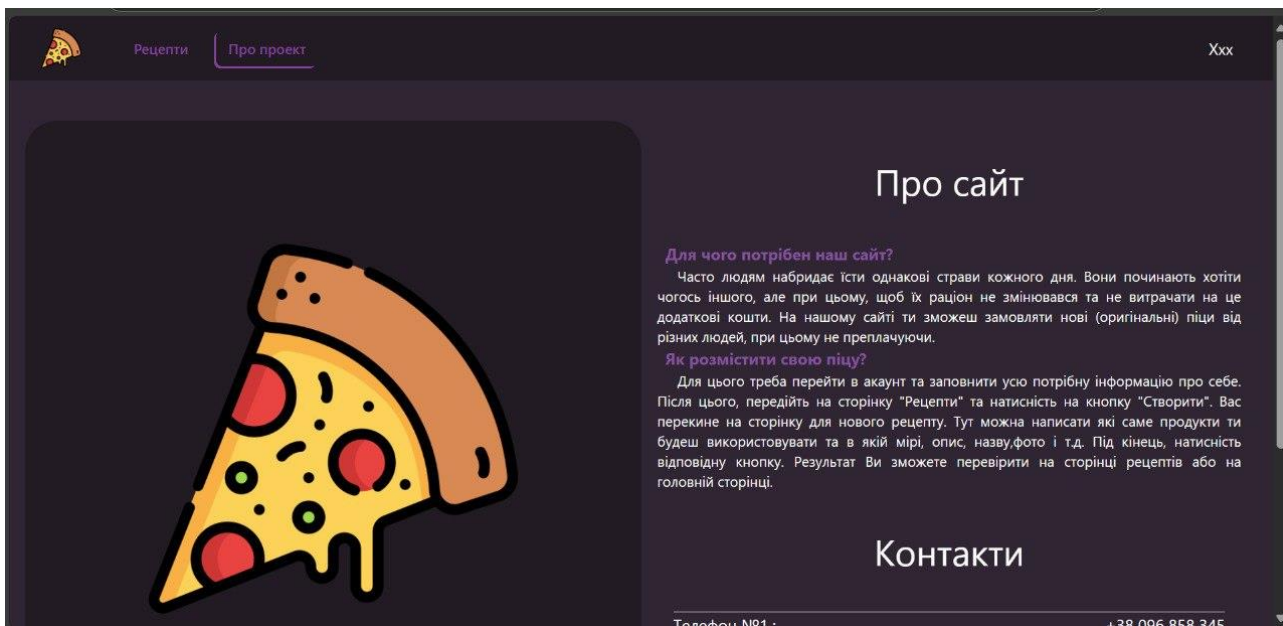
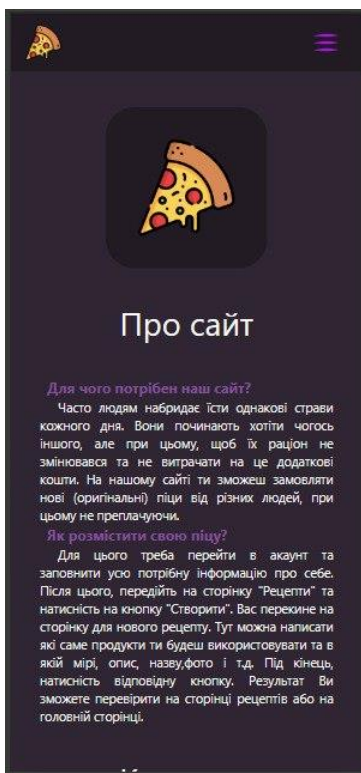
### 2. Мої замовлення



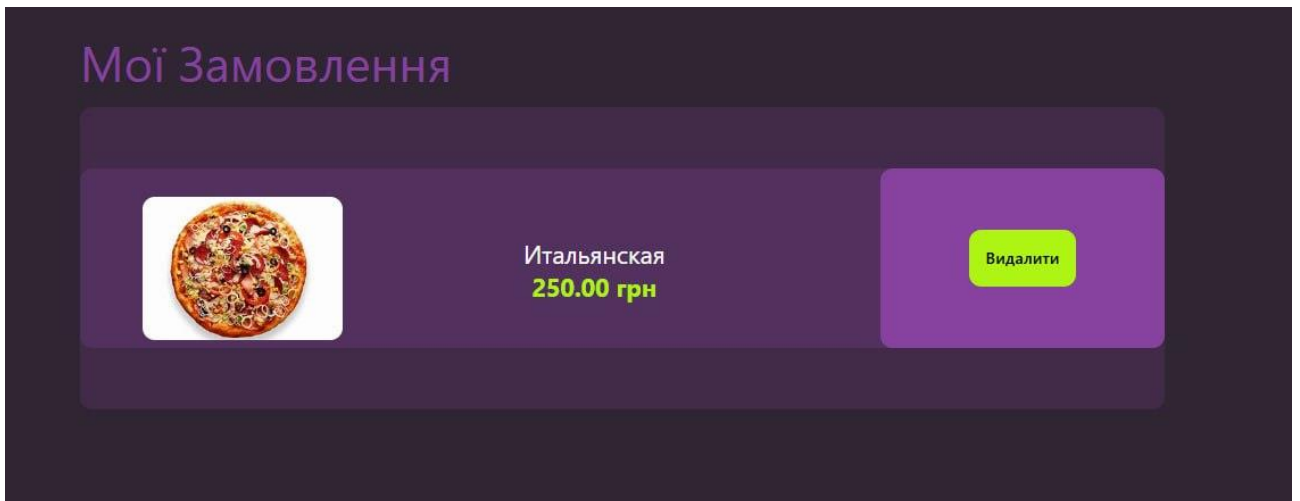
### 3. Панель адміністратора/ додавання рецепту/товару



### 4. Про нас



## 5. Кошик



## 6. Додавання товару

Фото

X Змінити

Назва рецепту

Назва рецепту

Ціна (грн)

0

Опис

Опишіть ваш рецепт

Зберегти

## 7. Вхід та реєстрація

Вхід

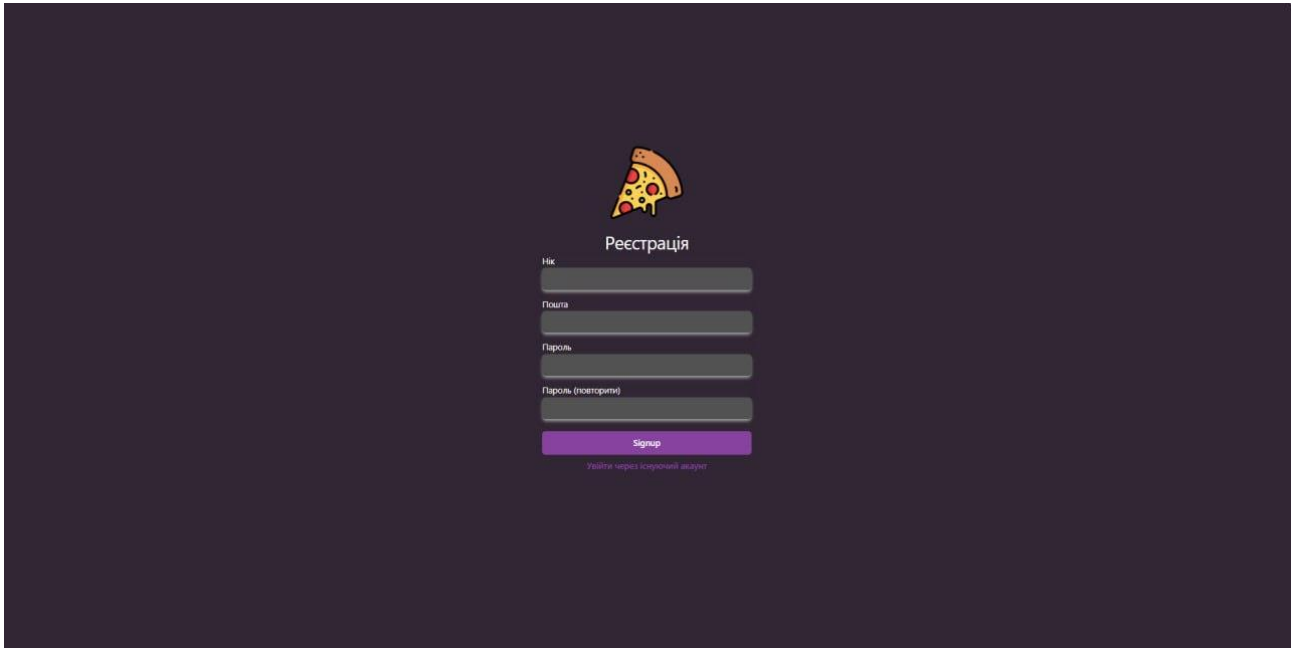
Пошта

Пароль

Запам'ятати мене

Вхід

Створити обліковий запис



Registration form on a dark background. At the top center is a cartoon pizza slice icon. Below it is the title "Реєстрація". The form consists of four input fields: "Ім'я", "Пошта", "Пароль", and "Пароль (пovторити)". Below the fields is a purple "Sign up" button. At the bottom, there is a small link: "Увійти через існуючий акаунт".

Реєстрація

Ім'я

Пошта

Пароль

Пароль (пovторити)

Sign up

[Увійти через існуючий акаунт](#)