

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту  
(повна назва)

Кафедра Інформатики  
(повна назва)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**Пояснювальна записка**

рівень вищої освіти перший (бакалаврський)

**РОЗРОБЛЕННЯ ВЕБЗАСТОСУНКУ ДЛЯ ПОШУКУ ТА**  
**ФОРМУВАННЯ ЗАМОВЛЕННЯ ПРОДУКТІВ ХАРЧУВАННЯ**  
(тема)

Виконав:  
здобувач 4 року навчання,  
групи ІТІНФ-21-3

Увохова І. С.  
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки  
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика  
(повна назва освітньої програми)

Керівник доц. Творошенко І. С.  
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики \_\_\_\_\_  
(підпис)

Кобилін О. А.  
(прізвище, ініціали)

2025 р.

## Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджментуКафедра ІнформатикиРівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки  
(код і повна назва)Тип програми освітньо-професійнаОсвітня програма Інформатика  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

«\_\_\_\_\_» \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУздобувачеві Увоховій Ірині Сергіївні  
(прізвище, ім'я, по батькові)1. Тема роботи Розроблення вебзастосунку для пошуку та формування замовлення продуктів харчування

затверджена наказом університету від 19 травня 2025 року № 381Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 22 травня 2025 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, інструмент адміністрування бази даних DBeaver, програмне забезпечення Rational Rose, програмне середовище розробки Microsoft Visual Studio Code.

4. Перелік питань, що потрібно опрацювати в роботі \_\_\_\_\_

1. Аналіз існуючих вебзастосунків для пошуку та формування замовлення продуктів харчування.2. Розроблення структури вебзастосунку для пошуку та формування замовлення продуктів харчування.3. Розроблення вебзастосунку для пошуку та формування замовлення продуктів харчування.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Мета роботи, об'єкт роботи, актуальність роботи, постановка задачі, моделювання бази даних, вибір інструментальних засобів, деталі розробки, тестові зображення.

---



---



---



---



---



---



---

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	07.04.2025	
2	Аналіз завдання, підбір літератури	08.04.25-10.04.25	
3	Аналіз літератури з досліджуваної проблеми	11.04.25-17.04.25	
4	Розроблення структури вебзастосунку	18.04.25-22.04.25	
5	Аналіз технічних засобів	23.04.25-25.04.25	
6	Програмна реалізація	26.04.25-11.05.25	
7	Оформлення пояснювальної записки	12.05.25-20.05.25	
8	Перевірка на нормоконтроль	21.05.25-01.06.25	
9	Перевірка на плагіат	21.05.25-01.06.25	
10	Рецензування	21.05.25-01.06.25	
11	Підготовка презентації та доповіді	21.05.25-18.06.25	
12	Занесення роботи в електронний архів	02.06.25-18.06.25	
13	Попередній захист кваліфікаційної роботи	02.06.25-18.06.25	

Дата видачі завдання 7 квітня 2025 р.

Здобувач \_\_\_\_\_  
(підпис)

Керівник роботи \_\_\_\_\_  
(підпис)

доц. Творошенко І. С.  
(посада, прізвище, ініціали)

## РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 67 с., 11 табл., 55 рис., 34 джерела.

ВЕБЗАСТОСУНОК, ПОШУК ПРОДУКТІВ ХАРЧУВАННЯ, ПОРІВНЯННЯ ЦІН, ФОРМУВАННЯ ЗАМОВЛЕННЯ, БАЗА ДАНИХ, СТАНДАРТ IDEF3, ДІАГРАМА КЛАСІВ, ДІАГРАМА ПРЕЦЕДЕНТІВ.

Об'єктом роботи є процес розроблення вебзастосунку для пошуку та формування замовлення товару.

Метою роботи є розроблення вебзастосунку для пошуку та формування замовлення продуктів харчування, що буде відповідати сучасним потребам користувачів і допомагати їм у виборі найвигідніших пропозицій.

У процесі роботи проаналізовано існуючі системи для пошуку товарів, літературні джерела та необхідні для розробки технології. Розроблено діаграми для опису структури вебзастосунку та бази даних.

У результаті роботи здійснена програмна реалізація вебзастосунку для пошуку та формування замовлення продуктів харчування.

WEB APPLICATION, FOOD SEARCH, PRICE COMPARISON, ORDER FORMATION, DATABASE, IDEF3 STANDARD, CLASS DIAGRAM, USE CASE DIAGRAM.

The object of the work is the process of developing a web application for searching and forming an order for goods.

The purpose of the work is to develop a web application for searching and forming an order for food products, which will meet the modern needs of users and help them choose the most profitable offers.

In the process of work, existing systems for searching for goods, literary sources and technologies necessary for development were analyzed. Diagrams were developed to describe the structure of the web application and database.

As a result of the work, a software implementation of a web application for searching and forming an order for food products was carried out.

## ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів .....	6
Вступ.....	7
1 Аналіз існуючих вебзастосунків для пошуку та формування замовлення продуктів харчування .....	8
1.1 Аналіз сучасних вебзастосунків для пошуку та формування замовлення продуктів харчування в Україні та за кордоном.....	8
1.2 Аналіз літературних джерел щодо апробації результатів розроблення вебзастосунків для пошуку та формування замовлення продуктів харчування .....	14
1.3 Постановка задачі .....	18
2 Розроблення структури вебзастосунку для пошуку та формування замовлення продуктів харчування.....	20
2.1 Особливості структури бази даних вебзастосунку для пошуку та формування замовлення продуктів харчування .....	20
2.2 Моделювання структури вебзастосунку для пошуку та формування замовлення продуктів харчування .....	29
3 Розроблення вебзастосунку для пошуку та формування замовлення продуктів харчування .....	40
3.1 Вибір інструментальних засобів для реалізації поставленої задачі ..	40
3.2 Етапи розроблення вебзастосунку для пошуку та формування замовлення продуктів харчування .....	42
3.2.1 Розроблення клієнтської частини вебзастосунку для пошуку та формування замовлення продуктів харчування .....	43
3.2.2 Розроблення серверної частини вебзастосунку для пошуку та формування замовлення продуктів харчування .....	49
3.3 Тестування розробленого вебзастосунку та аналіз результатів .....	53
3.4 Перспективи подальшої роботи .....	61
Висновки .....	63
Перелік джерел посилання .....	64

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних

IT – Information Technology (інформаційні технології)

ER-діаграма – Entity-Relationship діаграма (діаграма сутність-зв'язок)

СУБД – система управління базами даних

ПК – первинний ключ

ЗК – зовнішній ключ

IDEF3 – Integration DEFinition for Process Description Capture Method  
(інтегрований метод опису процесів)

UOB – Unit of Behavior (одиниця поведінки)

ELAB – Elaboration (уточнення)

URL – Uniform Resource Locator (уніфікований локатор ресурсу)

ПЗ – програмне забезпечення

VS Code – Visual Studio Code

SQL – Structured Query Language (мова структурних запитів)

Frontend – частина вебзастосунку, з якою безпосередньо взаємодіє користувач

Backend – частина вебзастосунку, яка працює на сервері

л – літр

г – грам

## ВСТУП

Сучасне життя показало, на скільки важливо мати можливість вирішувати буденні задачі дистанційно. Цифрові технології впевнено увійшли в життя людей: мобільний банкінг, цифрові черги, електронні щоденники та онлайн-послуги – все це стало невід’ємною частиною існування.

Проте, навіть із таким стрімким розвитком онлайн-сервісів та постійно зростаючою кількістю онлайн-магазинів, купівля продуктів харчування досі залишається незручною. Основною проблемою є порівняння цін на товари, адже для того, щоб знайти найдешевший варіант, доводиться відкривати одразу кілька сторінок та порівнювати вартість вручну. Як результат, люди втрачають багато часу і часто переплачують за покупки. Саме тому виникає потреба у створенні застосунку, який би дозволив шукати продукти харчування не у конкретному магазині, а порівнювати ціни серед різних торгових марок. Окрім вже описаного порівняння вартості, такий застосунок має включати в себе й інші важливі функції, такі як формування кошику та оформлення замовлення з можливістю подальшої оплати – в цьому випадку користувачі зможуть економити не тільки гроші, а й час, що є основною потребою сучасності.

Розробка такої системи вимагає не лише суто програмної реалізації, а й розуміння людської поведінки. Врахування потреб та пріоритетного для користувачів функціоналу є важливим для створення даної платформи.

Актуальність роботи полягає у необхідності створення універсального вебзастосунку, який би давав можливість безперешкодно шукати продукти харчування за різними критеріями, порівнювати їх вартість у різних магазинах та замовляти доставку заради економії зусиль, часу та грошей.

# 1 АНАЛІЗ ІСНУЮЧИХ ВЕБЗАСТОСУНКІВ ДЛЯ ПОШУКУ ТА ФОРМУВАННЯ ЗАМОВЛЕННЯ ПРОДУКТІВ ХАРЧУВАННЯ

1.1 Аналіз сучасних вебзастосунків для пошуку та формування замовлення продуктів харчування в Україні та за кордоном

У сучасному світі нікого не здивувати таким об'єктом, як магазин, навіть навпаки – можна здивувати їх кількістю. Із такою конкуренцією власникам бізнесів доводиться вигадувати різні способи, аби триматися «на плаву» та набирати популярність. Хтось працює над якістю, хтось – над кількістю. Зростають мережі, вигадуються нові акції, проте у сучасному суспільстві все більше виникає потреба у діджиталізації. Інтернет-сайти, Instagram-сторінки, акаунти у Facebook та Тік-Ток – це мінімум, який можна побачити у 2025 році. Великі мережі магазинів поширюють свою роботу через мобільні застосунки.

У 21 столітті, коли час вважається одним з найцінніших ресурсів, мобільні технології стали невід'ємною частиною повсякденного життя. Смартфони, які нещодавно використовувалися переважно для зв'язку, сьогодні стали універсальними інструментами для вирішення великої кількості задач, включаючи онлайн-шопінг. Мобільні застосунки для замовлення товарів пропонують зручність, доступність і широкий вибір, що значно полегшує та прискорює процес закупів. З ними покупцям не потрібно витрачати свій час на відвідування магазинів, пошуки потрібного товару та довгі черги. Все, що необхідно – це кілька кліків у мобільному застосунку, щоб знайти потрібний товар, порівняти ціни у різних продавців, прочитати відгуки і зробити замовлення. Однією з головних переваг таких онлайн-платформ є можливість купувати товари в будь-який час і з будь-якого місця. Незалежно від того, де є користувач застосунку: вдома, на роботі або в дорозі, він здійснить покупки тоді, коли це зручно саме йому. Це особливо

актуально в умовах сучасного міста, де панує шалений ритм життя та нерідко просто немає часу на традиційний шопінг.

Крім того, мобільні застосунки надають змогу отримувати найбільш актуальну інформацію про товари, акції та ціни. Завдяки цьому користувачі завжди можуть бути в курсі найбільш вигідних пропозицій і можуть заздалегідь планувати свої покупки. Інформація про наявність товарів у магазині, можливість порівняння цін та характеристик, велика кількість відгуків інших покупців – усе це робить процес вибору простішим і більш обґрунтованим. Ще одним важливим аспектом є зручність доставки. Більшість платформ пропонують різні варіанти, найпопулярнішими з яких є кур'єрська доставка та можливість самовивозу з найбільш зручного для користувача магазину. Це дозволяє покупцям обрати найзручніший спосіб отримання товарів саме для них.

Перші застосунки для шопінгу онлайн почали з'являтися на початку 2000-х років, разом із розвитком мобільних технологій і зростанням популярності інтернету. Проте, справжній прорив у цій сфері відбувся з появою iPhone у 2007 році та відкриттям App-Store у 2008 році. Це дало поштовх та змогу компаніям створювати більш зручні та функціональні програми. Одними із перших великих онлайн-ритейлерів, які запустили власний мобільний застосунок були Amazon та eBay (рис.1.1).

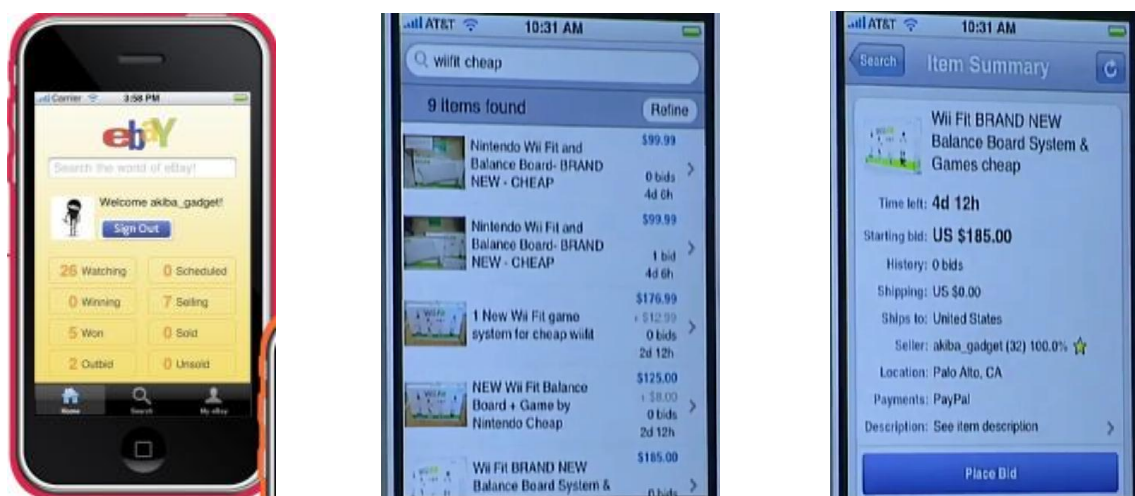


Рисунок 1.1 – Дизайн першого мобільного застосунку компанії eBay

Перші продукти Amazon та eBay з'явилися у 2008 році, вони дозволяли людям шукати та купувати товари безпосередньо зі своїх гаджетів. З цього часу і інші великі компанії почали активно розробляти концепти власних мобільних застосунків. Перші варіанти таких програм були досить простими та мали доволі обмежений функціонал, однак вони впевнено стали основою для подальшого розвитку цієї сфери. З часом вони еволюціонували, ставши більш зручними та багатофункціональними, що дозволило мобільним шопінг-платформам стати ключовим інструментом для сучасних споживачів.

Станом на 2025 рік ринок маркетплейсів налічує більше 100000 таких продуктів. Звичайно, це число включає в себе і глобальні платформи, відомі по всьому світу (Amazon, Alibaba, Walmart, eBay); і компанії поменше (Allegro, Temu, Shein, Rozetka, Makeup, Lidl, Aldi, Kaufland); і якісь регіональні застосунки (АТБ, Сільпо, Biedronka, Епіцентр, Żabka).

Із такою кількістю доступних магазинів, кожен із нас іноді задумується: «А де ж дешевше цей товар?». На сьогодні, застосунків, які порівнювали б між собою ціни з різних магазинів не так вже й багато.

Основним працюючим на сьогодні прикладом такої програми в Україні є Tabletki.ua [1]. Мета цього застосунку – допомогти користувачам знайти потрібні ліки, відсортувавши аптеки по відстані, вартості та наявності препарату, що дозволяє споживачу знайти найкращий варіант і зекономити сили, час та гроші. Також можна прочитати склад препаратів, знайти аналог і навіть скласти список ліків, щоб дізнатися про їх загальну вартість у конкретній аптеці (рис. 1.2). Проте, після користування даною платформою, окрім переваг, описаних вище, можна виділити також і кілька незначних недоліків. Основним недоліком є те, що у застосунку нема можливості сортування за кількома полями одночасно (наприклад ціна і відстань), натомість можна лише обрати радіус, в якому будуть запропоновані варіанти. Як недолік, також можна виділити і повну залежність від дистриб'юторів, тож, якщо аптека своєчасно не оновить дані про наявність товару, це призведе до певних незручностей і втрати часу.

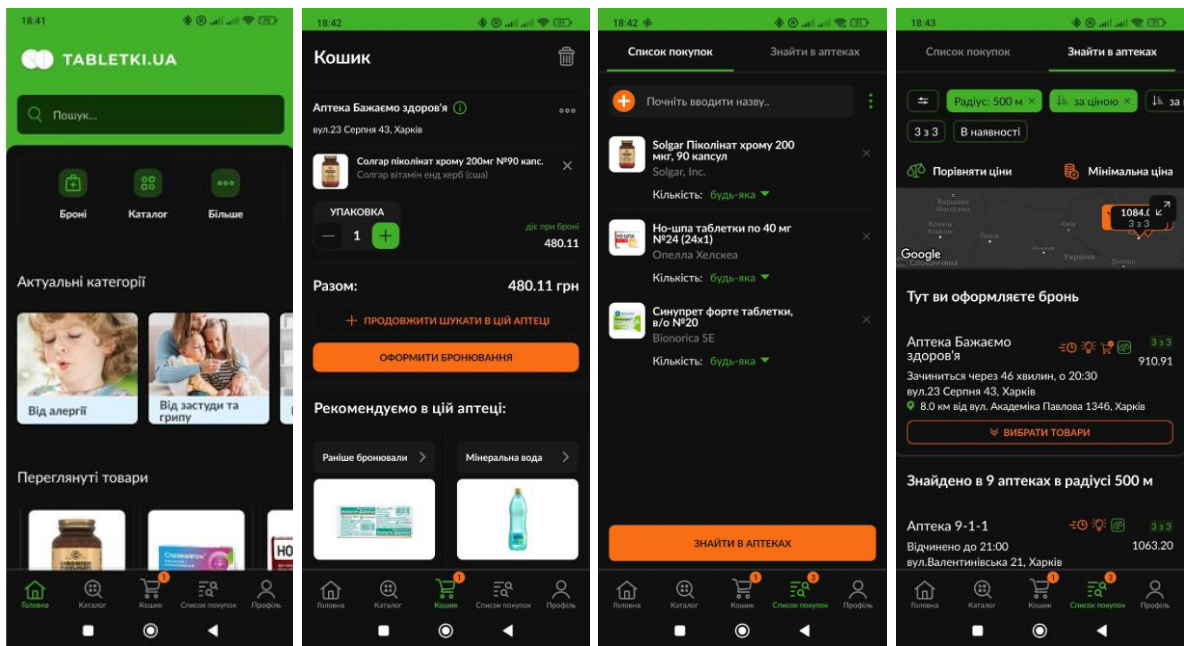


Рисунок 1.2 – Екранні форми застосунку Tabletki.ua

Крім того, оскільки застосунок не є інтернет-магазином, в ньому відсутня можливість замовлення доставки додому.

За кордоном аналогом є польський застосунок Ceneo [2]. На відміну від вітчизняної платформи, тут можна порівняти ціни на широкий спектр товарів, таких як одяг, техніка, ліки, продукти харчування, товари для дому та багато іншого (рис. 1.3).

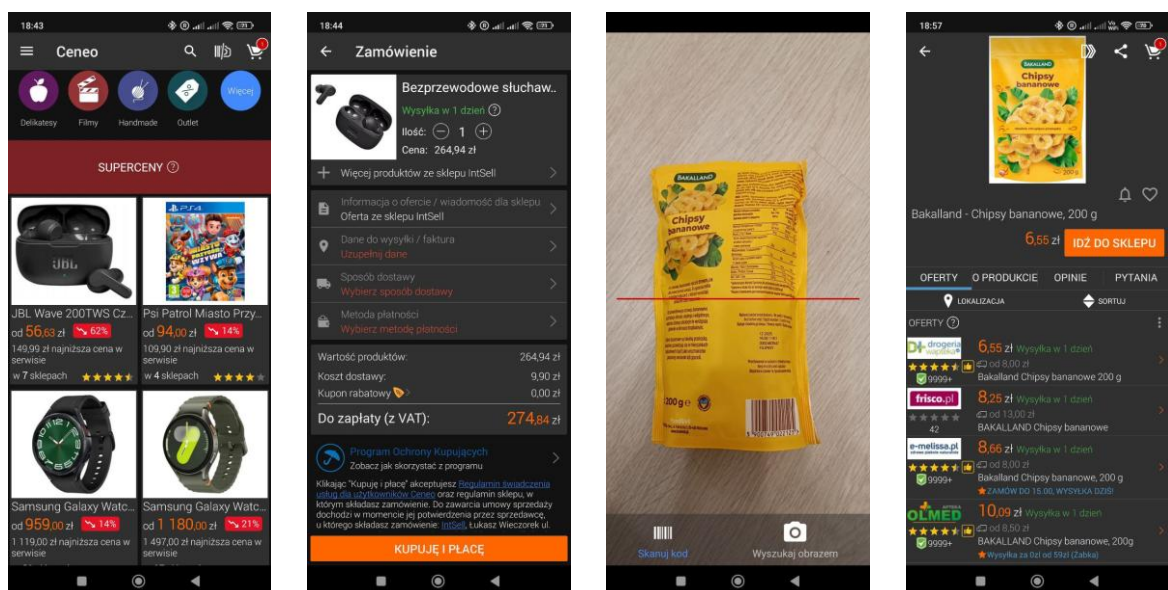


Рисунок 1.3 – Екранні форми застосунку Ceneo

Перевагою Сенео є можливість сканування штрих-коду товару або фотографування його образу, що у подальшому дозволяє з легкістю знайти товар, навіть не вводячи його назви.

Недоліком застосунку Сенео є те, що більшість запропонованих товарів порівнюються у інтернет-магазинах, таких як eBay, Allegro або Amazon, а не в статичних, тобто у більшості випадків немає можливості самовивозу, є лише доставка, що значно зменшує шанс знайти свіжі продукти, такі як овочі, фрукти або м'ясо. Також відсутня можливість знайти кілька товарів одночасно.

Ще одним прикладом такого сервісу є британський Trolley.co [3]. Як і у вище наведених аналогах, основною його функцією є пошук товарів у британських маркетах або онлайн-магазинах та порівняння цін. Тут також присутня можливість сканування штрих-коду, що робить пошук більш швидким (рис. 1.4).

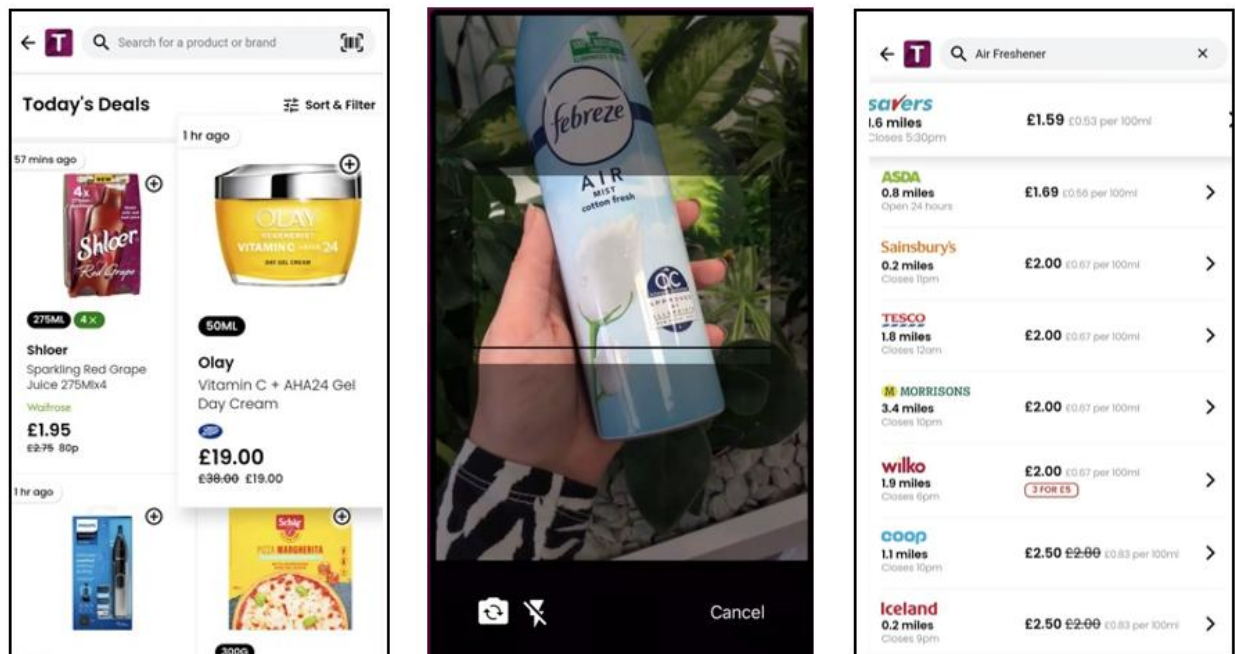


Рисунок 1.4 – Екранні форми застосунку Trolley.co

Прочитавши відгуки, із основних недоліків можна виділити відсутність деяких досить популярних товарів та магазинів у базі [4].

Із українських аналогів є також інтернет-сайт Price Navigator [5]. Він, як і польський застосунок Ceneo, пропонує користувачам порівнювати вартість товарів різного типу, від продуктів харчування до автомобілів (рис. 1.5).

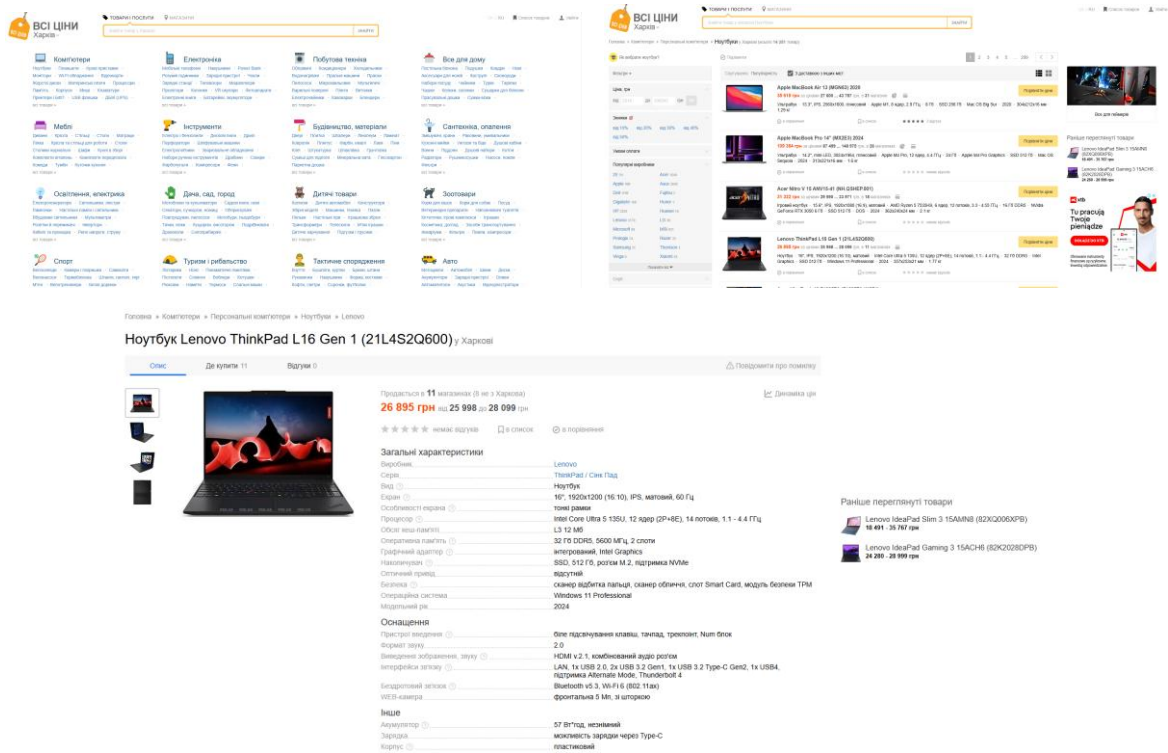


Рисунок 1.5 – Екранні форми сайту Price Navigator

Перевагою платформи Price Navigator є досить широкий функціонал та інтуїтивно зрозумілий дизайн. Із недоліків, аналогічно британській платформі Trolley.co, можна зазначити обмеженість бази магазинів та велику кількість реклами.

Загалом, мобільні застосунки стали справжньою революцією у світі шопінгу. Вони змінили людські звички, зробили процес покупок більш зручним та доступним.

У найближчому майбутньому можна очікувати, що всі ці технології продовжуватимуть розвиватися, надаючи споживачам ще більше можливостей для вигідних покупок.

## 1.2 Аналіз літературних джерел щодо апробації результатів розроблення вебзастосунків для пошуку та формування замовлення продуктів харчування

Розробка мобільних застосунків для порівняння цін на продукти харчування є досить актуальною темою в контексті розвитку інформаційних технологій. У зв'язку із постійно зростаючою популярністю онлайн-шопінгу, все більше виникає необхідність створення інструментів, які зможуть допомогти користувачам не лише знайти потрібний товар, але й обрати його за найбільш вигідною ціною. Це питання особливо актуальне для таких ринків, як продукти харчування, де ціни можуть суттєво варіюватися в залежності від пори року, регіону та магазину.

Аналіз літератури з цієї теми дозволяє виявити як загальні тенденції в розробці подібних застосунків, так і специфічні рішення, що можуть бути корисними для розробників. Більшість наукових робіт зосереджені на технічних аспектах створення таких платформ, зокрема на алгоритмах пошуку та порівняння цін, інтеграції з базами даних та забезпеченні актуальності інформації. Деякі дослідження також акцентують увагу на зручності користувацького інтерфейсу і функціональності, яка дозволяє користувачам отримувати не тільки ціни, але й додаткову інформацію про продукти, наприклад їх склад, екологічність або відгуки інших покупців.

Важливим аспектом є також і аналіз переваг та недоліків існуючих рішень, що дозволяє зрозуміти шляхи подальшого вдосконалення. Наприклад, деякі дослідження підкреслюють важливість регулярного оновлення та адаптації застосунків до умов різних ринків, тоді як інші говорять про необхідність розширення функціоналу або інтеграцію з іншими сервісами.

У даній роботі буде розглянуто ключові наукові джерела, які аналізують різні аспекти розробки мобільних застосунків для порівняння цін

на продукти харчування, а також визначають перспективи та виклики в цій сфері.

У статті [6] аналізується зростання популярності мобільних застосунків для порівняння цін та їх вплив на сучасний ринок споживчих товарів. Основною метою є підкреслення важливості та переваг таких платформ для покупців, які прагнуть знайти найвигідніші пропозиції на ринку. Окрім цього, автором розглядається, як ці інструментальні засоби працюють, включаючи процес збору та аналізу інформації, а також алгоритми для відображення найкращих пропозицій. Окремо описується важливість інноваційних технологій, які дозволяють застосункам ефективно виконувати свої функції. Також досліджуються ринкові тенденції і прогнозується подальший розвиток мобільних платформ для порівняння цін, зазначаючи, що їх популярність буде зростати завдяки постійним технологічним вдосконаленням та змінам у споживацьких вподобаннях. Отже, спираючись на це джерело, можна зрозуміти, що проблематика даної роботи є досить актуальною.

У роботі [7] автори надають всебічний огляд аспектів, що стосуються розробки мобільних застосунків для порівняння цін. Вони починають з аналізу важливих функцій, які повинні бути реалізовані в системах такого типу (наприклад інтеграція з різними джерелами для отримання актуальних даних, можливість фільтрації та сортування товарів, а також надання детальної інформації про кожен продукт). Акцентується увага також і на користувацькому інтерфейсі: для забезпечення зручності та ефективності користування застосунком, інтерфейс повинен бути простим та інтуїтивно зрозумілим. Окрім цього, у даній статті розглядаються технології і платформи, які використовуються для розробки таких систем, описуються різні методи збору і обробки даних. У висновку до статті зазначено, що успішна розробка застосунку для порівняння цін вимагає не лише технічних знань, але й розуміння потреб користувачів і ринкових тенденцій. Проаналізувавши цю роботу можна знайти багато корисної інформації для

ІТ-спеціалістів, які тільки починають свій шлях із розробки мобільних платформ.

Статті [8, 9] описують безпосередньо розробку та реалізацію вебсайту для порівняння цін. Автори детально розглядають процес пошуку та аналізу цінової інформації з різних інтернет-магазинів, використовуючи методи вебскрейпінгу [10] для автоматизації збору даних. У роботі висвітлюються ключові технічні аспекти, такі як побудова архітектури вебсайту, створення алгоритмів для ефективного отримання та обробки даних, а також забезпечення актуальності та точності інформації. Результати дослідження демонструють ефективність запропонованого підходу для створення вебсайту, який дозволяє користувачам порівнювати ціни на різні товари, тим самим допомагаючи їм приймати більш обґрунтовані рішення щодо покупок. Стаття підсумовує, що використання вебскрейпінгу в поєднанні з правильною архітектурою вебсайту може стати потужним інструментом для підвищення конкурентоспроможності на ринку.

У науковій роботі [11] проводиться аналіз особливостей поведінки споживачів при купівлі продуктів харчування в інтернеті. Це дослідження враховує вік і купівельні переваги покупців онлайн-магазинів. Результати тестування показують, що існує зв'язок між віком клієнтів та їх онлайн-покупками. Зауважено, що покупці, які купують продукти в інтернет-магазині, все ще неохоче купують швидкопсувні товари (наприклад м'ясо або випічку) і віддають перевагу товарам, упакованим безпосередньо виробником. Дослідження дає можливість розуміння клієнтів, купівельної поведінки в цілому та їх потреби щодо якості обслуговування, умов оплати і доставки. Також можна розглянути і вплив таких застосунків на бізнес.

У статті [12] досліджується вплив вебсайтів для порівняння цін на функціонування ринку та його структуру. Автори аналізують, як ціни, конкуренція та загальна ефективність роботи ринку змінюються під впливом таких платформ. Стаття підкреслює, що порівняльні сайти сприяють підвищенню прозорості ринку, полегшуючи споживачам доступ до

інформації. Це, в свою чергу, може збільшити конкуренцію між продавцями, змушуючи їх знижувати ціни та підвищувати якість обслуговування.

Окрім статей про застосунки та вебсайти для порівняння цін, корисними є також джерела про створення платформ для замовлення доставки товарів.

У роботі [13] описується процес розробки мобільного застосунку для доставки продуктів на замовлення. Мета проекту полягала в спрощенні процесу покупок у супермаркетах через інтуїтивний інтерфейс. Результатом роботи стало створення функціональної платформи, що включає простий пошук товарів та легкий процес оформлення замовлення. Стаття детально описує етапи дослідження користувачів, дизайн інтерфейсу та технічну реалізацію, надаючи корисні поради для дизайнерів і розробників.

Метою дослідження [14] є вивчення того, як використання мобільних пристроїв у магазинах впливає на рішення про покупки, споживацькі звички та загальний досвід клієнтів. Дослідження прагне виявити конкретні аспекти, в яких смартфони можуть як покращити, так і ускладнити процес закупів в продуктових магазинах. Під час аналізу автори використовують дані, зібрані через опитування споживачів, і результати спостереження за поведінкою покупців у супермаркетах. Розглядаються різні сценарії використання мобільних пристроїв, включаючи пошук інформації про продукти, порівняння цін і використання мобільних купонів. Результати дослідження вказують на те, що використання смартфонів позитивно впливає на деякі аспекти споживацького досвіду. Мобільні пристрої також сприяють більшій зручності і швидкості покупок. Однак, спостерігаються й деякі негативні ефекти, зокрема можливість відволікання і підвищення конкуренції серед продавців за увагу споживачів. Стаття підкреслює важливість адаптації роздрібних магазинів до нових технологій. Мережі повинні враховувати вплив мобільних застосунків на покупців і розвивати стратегії для інтеграції цих технологій у свої процеси.

Підсумовуючи аналіз літературних джерел [6–14], можна зробити висновок про те, що наразі, зважаючи на шалений ритм життя і доволі стрімкий розвиток мобільних технологій, дуже важливо використовувати у роботі сучасні методи та інноваційні інструменти. Власники магазинних бізнесів – не виключення.

Мобільні платформи для порівняння цін та замовлення товарів збільшують конкуренцію, одночасно полегшують і ускладнюють багато процесів в інформаційній системі. Проте, користь та актуальність таких застосунків для споживачів очевидна.

### 1.3 Постановка задачі

У теперішніх реаліях, коли інфляція зростає в усьому світі, вільного часу, здається, меншає з кожним днем, а стрес від походу в магазин стає буденністю, саме вебзастосунки для пошуку та формування замовлення продуктів харчування стають вірними помічниками споживачів, адже завдяки ним можна економити гроші та час, планувати свої покупки та більше зосереджуватися на важливих моментах життя.

Об'єктом роботи є процес розроблення вебзастосунку для пошуку та формування замовлення товару.

Метою роботи є розроблення вебзастосунку для пошуку та формування замовлення продуктів харчування, що буде відповідати сучасним потребам користувачів і допомагати їм у виборі найвигідніших пропозицій.

Для досягнення мети необхідно вирішити такі завдання:

- проаналізувати літературні джерела щодо застосунків такого типу;
- змодельювати та створити базу даних;
- змодельювати структуру вебзастосунку;
- обрати середовище розробки та інструментальні засоби;

- розробити вебзастосунок для порівняння цін, пошуку та формування замовлення продуктів харчування;
- провести тестування застосунку;
- виявити перспективи подальшої роботи;
- проаналізувати пророблену роботу і зробити висновки.

## 2 РОЗРОБЛЕННЯ СТРУКТУРИ ВЕБЗАСТОСУНКУ ДЛЯ ПОШУКУ ТА ФОРМУВАННЯ ЗАМОВЛЕННЯ ПРОДУКТІВ ХАРЧУВАННЯ

2.1 Особливості структури бази даних вебзастосунок для пошуку та формування замовлення продуктів харчування

Одним із ключових компонентів інформаційного вебзастосунок є база даних, функцією якої є забезпечення збереження, оброблення та зручного управління інформацією. Ретельно продумана структура БД не лише гарантує цілісність важливих даних, але й забезпечує високу швидкість доступу до них, що є важливим фактором для ефективного керування системою.

У цьому розділі, за допомогою ER-діаграми, будуть візуалізовані основні сутності БД, їх атрибути, зв'язки, а також загальна модель, що є в основі розроблення вебзастосунок для пошуку та формування замовлення продуктів харчування.

У контексті даної теми база даних використовується для збереження усієї інформації про користувачів, товари, магазини, замовлення та інші пов'язані об'єкти. Її проєктування відбувалося з урахуванням функціональних вимог до платформи, таких як пошук продуктів харчування за різними параметрами, відображення актуальної вартості, формування кошиків, а також зберігання історії замовлень.

Для розробки БД та створення ER-діаграми (рис. 2.1) використано:

- DBeaver [15] – програма для моделювання діаграм, що дозволяє наочно показати усю внутрішню структуру системи;
- SQLite [16] – СУБД для зберігання інформації;
- SQLite Studio [17] – графічне середовище для більш зручної роботи із запитам.

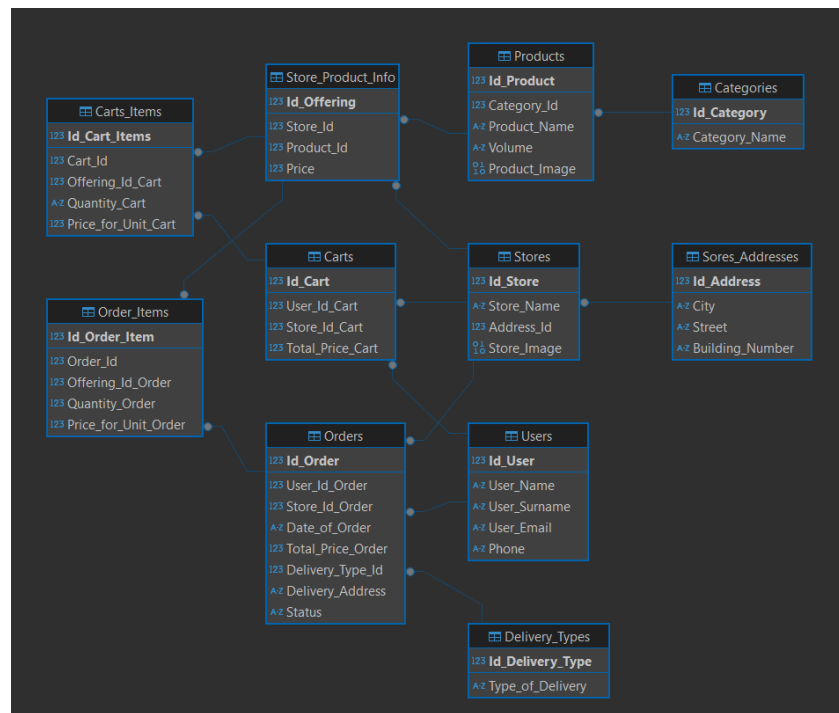


Рисунок 2.1 – ER-діаграма БД вебзастосунку для пошуку та формування замовлення продуктів харчування

Переваги програми DBeaver:

- підтримка багатьох СУБД: DBeaver працює із багатьма найпопулярнішими СУБД різних типів, як реляційними, такими як SQLite, MySQL, Oracle, SQLServer, PostgreSQL тощо, так і нереляційними;
- кросплатформність: DBeaver легко встановлюється та безперерійно працює на основних операційних системах, таких як Windows, Linux та macOS;
- безкоштовна: DBeaver має безкоштовну Community версію, інструментарію якої вистачає для легкого моделювання структури БД та управління нею;
- зручний та інтуїтивно-зрозумілий графічний інтерфейс: DBeaver має досить простий та зрозумілий інтерфейс, завдяки чому в ній можна працювати навіть без спеціальної підготовки.

Переваги СУБД SQLite:

- проста інтеграція: SQLite не потребує встановлення окремого серверу, натомість зберігається звичайним файлом, який легко підключити;

– займає мало місця: сама програма SQLite та створювані у ній таблиці мають «легку вагу», що дозволяє використовувати її навіть в умовах обмеженості пам'яті;

– кросплатформність: SQLite працює на усіх основних операційних системах, таких як Windows, Linux, iOS, Android тощо;

– безкоштовність: SQLite є повністю безкоштовною та має відкритий код;

– портативність: БД зберігається єдиним файлом у форматі, що значно спрощує процес його перенесення або копіювання;

– надійність: SQLite є неодноразово протестованою СУБД, що підтверджує стабільність її роботи та збереження безпеки даних навіть при надзвичайних ситуаціях.

На ER-діаграмі представлені одинадцять сутностей:

– Users – таблиця зберігає дані про усіх користувачів вебзастосунку (табл. 2.1);

– Categories – таблиця містить інформацію про категорії продуктів харчування (табл. 2.2);

– Products – таблиця зберігає інформацію про товари (табл. 2.3);

– Stores\_Addresses – таблиця містить адреси магазинів (табл. 2.4);

– Stores – таблиця містить відомості про магазини, які співпрацюють із вебзастосунком (табл. 2.5);

– Store\_Product\_Info – таблиця зберігає актуальну інформацію щодо цін товарів у магазинах (табл. 2.6);

– Carts – таблиця містить перелік кошиків, що зареєстровані у системі (табл. 2.7);

– Cart\_Items – таблиця зберігає додані у кошик товари (табл. 2.8);

– Orders – таблиця формує список із вже оформлених замовлень (табл. 2.9);

– Delivery\_Types – таблиця містить різновиди доставки (табл. 2.10);

– Order\_Items – таблиця зберігає список замовлених товарів відповідно до id замовлення (табл. 2.11).

Таблиця 2.1 – Users вебзастосунку для пошуку та формування замовлення продуктів харчування

Назва поля	Тип даних	Null/Not null	Опис поля	Приклад
Id_User	INTEGER	Not null	Унікальний ідентифікатор користувача, ПК	25
User_Name	TEXT(30)	Not null	Ім'я користувача	Ірина
User_Surname	TEXT(30)	Not null	Прізвище користувача	Увохова
User_Email	TEXT(30)	Not null	Електронна пошта користувача	iryuna@gmail.com
Phone	TEXT(13)	Not null	Номер телефону користувача	+380961234567

Таблиця 2.2 – Categories вебзастосунку для пошуку та формування замовлення продуктів харчування

Назва поля	Тип даних	Null/Not null	Опис поля	Приклад
Id_Category	INTEGER	Not null	Унікальний ідентифікатор категорії, ПК	1
Category_Name	TEXT(20)	Not null	Ім'я користувача	Напої

Таблиця 2.3 – Products вебзастосунку для пошуку та формування замовлення продуктів харчування

Назва поля	Тип даних	Null/Not null	Опис поля	Приклад
Id_Product	INTEGER	Not null	Унікальний ідентифікатор товару, ПК	3
Category_Id	INTEGER	Not null	Ідентифікатор категорії, ЗК	2 (Солодощі)
Product_Name	TEXT(50)	Not null	Назва товару	Печиво «Ювілейний букет»
Volume	TEXT(10)	Not null	Об'єм	200 г
Product_Image	BLOB	Not null	Зображення товару	/9j/4AAQSkZJRgA...

Таблиця 2.4 – Stores\_Addresses вебзастосунку для пошуку та формування замовлення продуктів харчування

Назва поля	Тип даних	Null/Not null	Опис поля	Приклад
Id_Address	INTEGER	Not null	Унікальний ідентифікатор адреси, ПК	5
City	TEXT(50)	Not null	Місто	Харків
Street	TEXT(100)	Not null	Вулиця	Валентинівська
Building_Number	TEXT(10)	Not null	Номер будівлі	18Б

Таблиця 2.5 – Stores вебзастосунку для пошуку та формування замовлення продуктів харчування

<b>Назва поля</b>	<b>Тип даних</b>	<b>Null/Not null</b>	<b>Опис поля</b>	<b>Приклад</b>
Id_Store	INTEGER	Not null	Унікальний ідентифікатор магазину, ПК	3
Store_Name	TEXT(50)	Not null	Назва магазину	Посад
Address_Id	INTEGER	Not null	Ідентифікатор адреси магазину, ЗК	3 (Харків, Валентинівська, 18 Б)
Store_Image	BLOB	Not null	Зображення магазину	iVBORw0KGg...

Таблиця 2.6 – Store\_Product\_Info вебзастосунку для пошуку та формування замовлення продуктів харчування

<b>Назва поля</b>	<b>Тип даних</b>	<b>Null/Not null</b>	<b>Опис поля</b>	<b>Приклад</b>
Id_Offering	INTEGER	Not null	Унікальний ідентифікатор пропозиції, ПК	2
Store_Id	INTEGER	Not null	Ідентифікатор магазину, ЗК	4 (Сільпо)
Product_Id	INTEGER	Not null	Ідентифікатор товару, ЗК	1 (Coca-Cola, 0,5 л)
Price	REAL	Not null	Вартість товару у магазині	24,99

Таблиця 2.7 – Carts вебзастосунку для пошуку та формування замовлення продуктів харчування

Назва поля	Тип даних	Null/Not null	Опис поля	Приклад
Id_Cart	INTEGER	Not null	Унікальний ідентифікатор кошика, ПК	3
User_Id_Cart	INTEGER	Not null	Ідентифікатор користувача, ЗК	5
Store_Id_Cart	INTEGER	Not null	Ідентифікатор магазину, ЗК	4 (Сільпо)
Total_Price_Cart	REAL	Not null	Повна вартість кошика	380,59

Таблиця 2.8 – Cart\_Items вебзастосунку для пошуку та формування замовлення продуктів харчування

Назва поля	Тип даних	Null/Not null	Опис поля	Приклад
Id_Cart_Item	INTEGER	Not null	Унікальний ідентифікатор елемента кошика, ПК	2
Cart_Id	INTEGER	Not null	Ідентифікатор кошика, ЗК	7
Offering_Id_Cart	INTEGER	Not null	Ідентифікатор пропозиції, ЗК	9 (Кріп, Посад...)
Quantity_Cart	INTEGER	Not null	Кількість одиниць	1
Price_for_Unit_Cart	REAL	Not null	Вартість за одиницю товару	17,95

Таблиця 2.9 – Orders вебзастосунку для пошуку та формування замовлення продуктів харчування

<b>Назва поля</b>	<b>Тип даних</b>	<b>Null/Not null</b>	<b>Опис поля</b>	<b>Приклад</b>
Id_Order	INTEGER	Not null	Унікальний ідентифікатор кошика, ПК	3
User_Id_Order	INTEGER	Not null	Ідентифікатор користувача, ЗК	5
Store_Id_Order	INTEGER	Not null	Ідентифікатор магазину, ЗК	4 (Сільпо)
Date_of_Order	TEXT(20)	Not null	Дата замовлення	2025.04.15
Total_Price_Order	REAL	Not null	Повна вартість замовлення	380,59
Delivery_Type_Id	INTEGER	Not null	Ідентифікатор типу доставки (самовивіз / кур'єр), ЗК	Кур'єр
Delivery_Address	TEXT(100)	Null	Адреса доставки	Вул. Ак. Павлова, 132
Type_of_Payment	TEXT(30)	Not null	Тип оплати	Готівкою
Status	TEXT(10)	Not null	Статус замовлення	У процесі

Таблиця 2.10 – Delivery\_Types вебзастосунку для пошуку та формування замовлення продуктів харчування

Назва поля	Тип даних	Null/Not null	Опис поля	Приклад
Id_Delivery_Type	INTEGER	Not null	Унікальний ідентифікатор типу доставки, ПК	1
Type_of_Delivery	TEXT(30)	Not null	Тип доставки	Кур'єр

Таблиця 2.11 – Order\_Items вебзастосунку для пошуку та формування замовлення продуктів харчування

Назва поля	Тип даних	Null/Not null	Опис поля	Приклад
Id_Order_Item	INTEGER	Not null	Унікальний ідентифікатор елемента замовлення, ПК	5
Order_Id	INTEGER	Not null	Ідентифікатор замовлення, ЗК	8
Offering_Id_Order	INTEGER	Not null	Ідентифікатор пропозиції, ЗК	9 (Кріп, Посад...)
Quantity_Order	INTEGER	Not null	Кількість одиниць	1
Price_for_Unit_Order	REAL	Not null	Вартість за одиницю товару	17,95

База даних відіграє одну із головних ролей у роботі вебзастосунку для пошуку та формування замовлення продуктів харчування, тож його функціонування на пряму залежить від коректності побудови структури БД.

Отже, надана структура БД відповідає усім вимогам, що необхідні для створення потрібного функціоналу системи та зберігає у собі усі необхідні дані.

## 2.2 Моделювання структури вебзастосунку для пошуку та формування замовлення продуктів харчування

Моделювання структури вебзастосунку – це найголовніший етап розробки. Саме від цього процесу залежить, на скільки зручним, логічно правильним та працездатним буде результат.

Моделювання структури може бути різним, починаючи описом функціоналу словами, закінчуючи складними діаграмами із поясненнями для кожного елемента. В даній роботі структура вебзастосунку для пошуку та формування замовлення продуктів харчування буде представлена у вигляді діаграми IDEF3, діаграми класів та діаграми прецедентів.

Переваги вибору діаграми IDEF3:

- легкість сприйняття: діаграми складаються із простих, зрозумілих усім, елементів: блоків та стрілок. За рахунок цього, вони легко читаються та не потребують додаткових технічних знань;

- логічність та продуманість: діаграми IDEF3 будуються в такий спосіб, щоб закласти логіку процесів, що будуть проходити у системі, їх послідовність або паралельність. Саме завдяки цьому ця діаграма є максимально інформативною для програмістів;

- ієрархічна структура: діаграма будується у кілька етапів, складні процеси розбиваються на підпроцеси, що дозволяє не перенавантажувати діаграму і докладніше описувати процеси.

Основними елементами діаграм стандарту IDEF3 є:

– одиниці робіт (процеси) – центральні елементи системи, зображуються прямокутниками;

– зв'язки (потоки даних) – елементи системи, що зображуються стрілками та показують взаємовідношення робіт:

1) старша стрілка – зображується суцільною лінією, зв'язує одиниці робіт за принципом «Робота-джерело повинна закінчитися перш, ніж робота-мета розпочнеться»;

2) стрілка відношень – зображується пунктирною лінією, зв'язує одиниці робіт за принципом «Робота-джерело не обов'язково має закінчитися перш, ніж робота-мета розпочнеться», а також одиниці робіт з об'єктами посилянь;

3) потоки – зображується стрілкою з двома наконечниками, застосовується для опису факту використання одного об'єкту в двох, або більше одиницях робіт;

– перехрестя – елементи системи, що відображають логіку взаємодії за допомогою злиття та розгалуження стрілок:

1) асинхронне «І» – усі попередні/наступні процеси мають бути завершені/запущені;

2) асинхронне «Або» – один або кілька попередніх/наступних процесів мають бути завершені/запущені;

3) синхронне «І» – усі попередні/наступні процеси мають бути завершені/запущені одночасно;

4) синхронне «Або» – один або кілька попередніх/наступних процесів мають бути завершені/запущені одночасно;

5) виключне «Або» – тільки один попередній/наступний процес має бути завершений/запущений;

– об'єкти посилянь – елементи системи, що використовуються для зображення інформації, яку не можна пов'язати з іншими елементами системи, для залучення уваги користувача:

- 1) Object – використовується для опису факту, що в дії бере участь об'єкт, який заслуговує на окрему увагу;
- 2) GOTO – використовується для реалізації циклічності виконання дій;
- 3) UOB – використовується для багаторазового відображення дії, але без циклу;
- 4) Note – використовується для документування важливої інформації загального характеру;
- 5) ELAB – використовується для уточнення або більш докладного опису зображуваного на діаграмі [18].

Переваги вибору діаграми класів:

- наочність структури системи: діаграма класів демонструє усі класи, з яких складається система, їх атрибути, методи та конкретні типи зв'язків, що в майбутньому полегшує процес розуміння;
- зручна для документування: діаграму можна використовувати в технічних документах, оскільки вона без зайвих деталей описує загальну архітектуру системи;
- чіткий розподіл: діаграма дозволяє наочно побачити за що відповідає кожен конкретний клас, що допомагає уникнути повторень та змішування логіки.

На діаграмі кожен клас графічно представляє собою прямокутник, в якому описані ім'я класу, його атрибути та операції. Ці класи можуть поєднуватися між собою стрілками відношень. Існує 3 типи відношень:

- узагальнення (спадкування) – зв'язок, при якому об'єкт дочірнього елемента (нащадок) може використовуватися замість об'єкта батьківського елемента (предка). При цьому зв'язку «нащадок» має як властивості «предка», так і свої власні;
- залежність – відношення між двома сутностями, при якому зміна однієї з них, незалежної, може вплинути на семантику іншої, залежної;

– асоціація – двосторонній зв'язок між класами, який описує сукупність зв'язків. Різновидом асоціації є агрегація – зв'язок, що описує відношення «частина – ціле» [18].

Переваги діаграми прецедентів:

– інтуїтивно-зрозуміла: діаграма прецедентів, за допомогою простих елементів, наочно показує ролі, що існують у системі, та процеси, які вони виконують;

– є основою для створення інших діаграм: оскільки у діаграмі прецедентів описуються усі основні ролі та процеси, то вона може слугувати як основний шаблон для створення інших моделей та розуміння, який функціонал планується розробляти;

– допомагає виявити прогалини у вимогах: під час створення діаграми можуть виявитися надлишкові або відсутні варіанти використання, які, на етапі моделювання, ще можна прибрати або додати.

Основні елементи діаграми прецедентів:

– актор (людина, об'єкт, автоматизована система) – зображується у вигляді «людини», від якої ідуть стрілки до прецедентів;

– прецедент (варіант використання) – зображується у вигляді овалу.

У даній роботі діаграма IDEF3 буде створена у застосунку Erwin Process Modeler, а діаграми класів та прецедентів – у програмі Rational Rose. Ці програмні забезпечення мають інтуїтивно зрозумілий інтерфейс, а також увесь необхідний для моделювання діаграм інструментарій.

За допомогою діаграм стандарту IDEF3 (рис. 2.2 – рис. 2.8) буде описано процес реєстрації, пошуку та формування замовлення продуктів харчування у вебзастосунку зі сторони користувача.

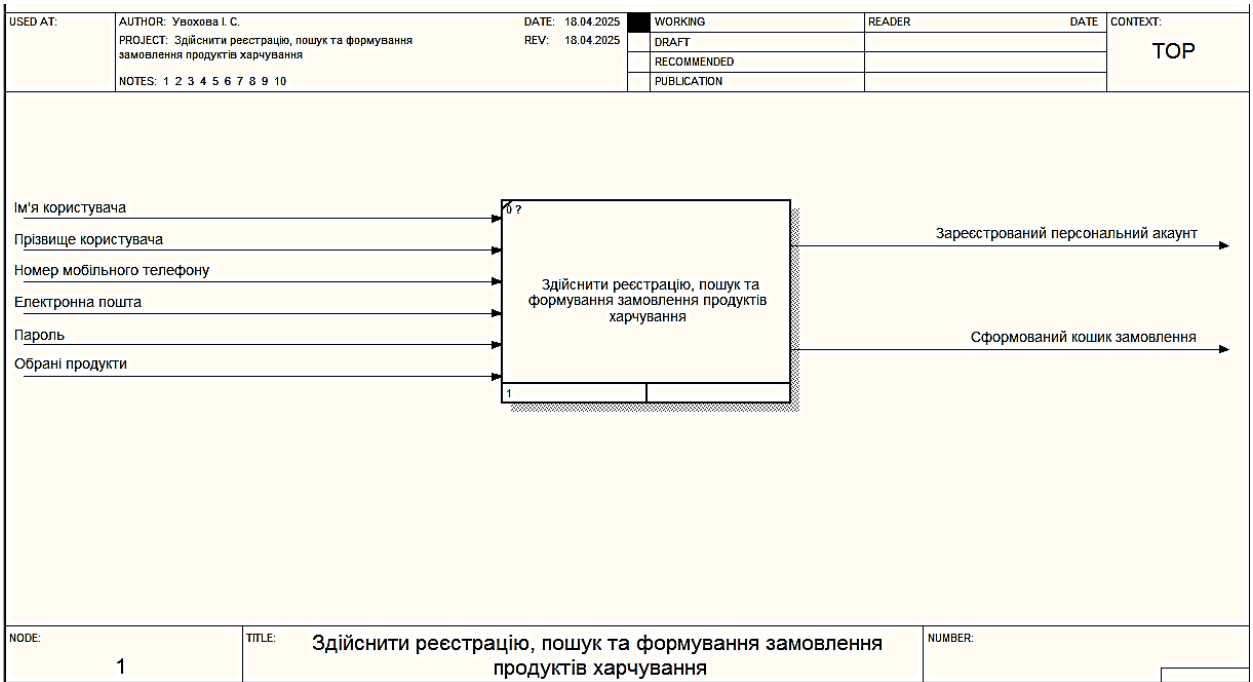


Рисунок 2.2 – Контекстна діаграма процесу «Здійснити реєстрацію, пошук та формування замовлення продуктів харчування» у нотації IDEF3

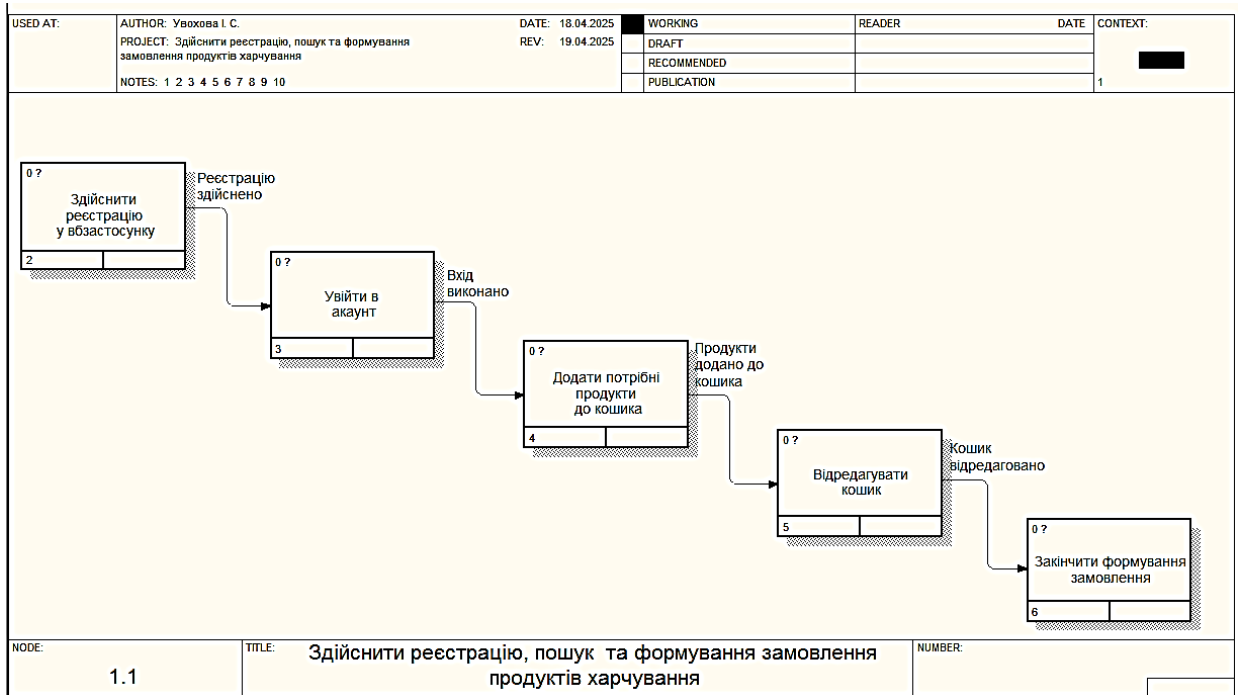


Рисунок 2.3 – Діаграма декомпозиції першого рівня процесу «Здійснити реєстрацію, пошук та формування замовлення продуктів харчування»



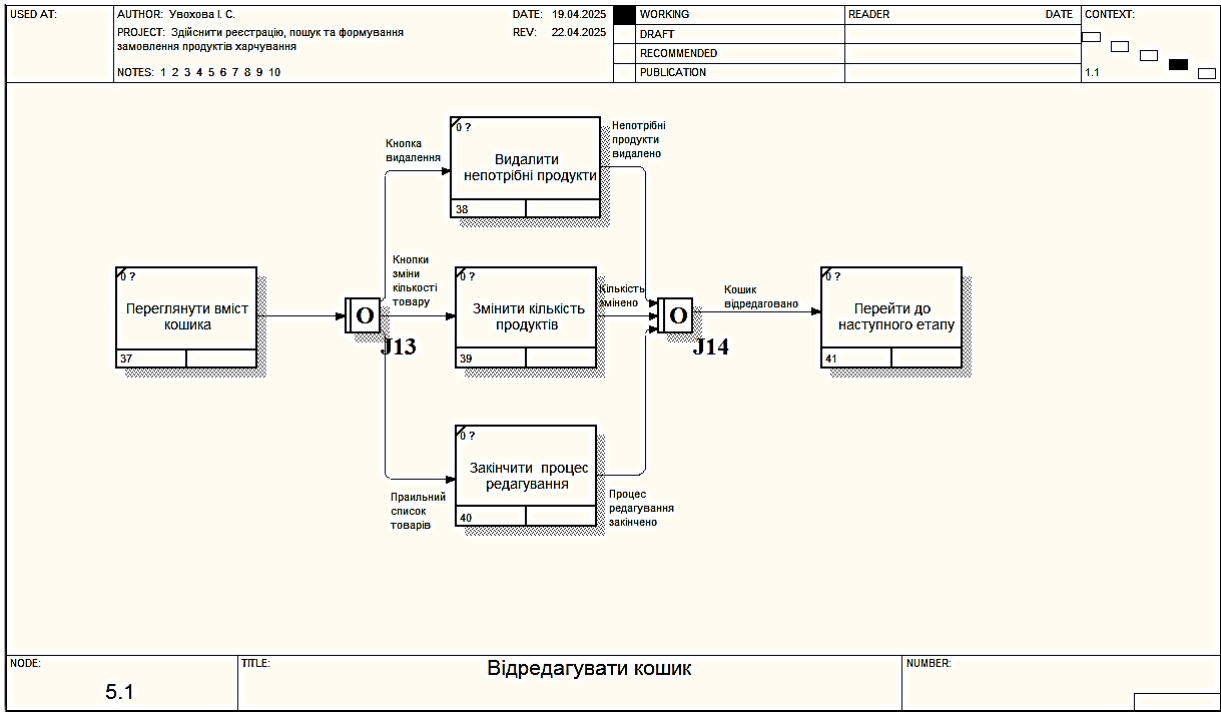


Рисунок 2.7 – Діаграма декомпозиції другого рівня процесу «Відредагувати кошик»

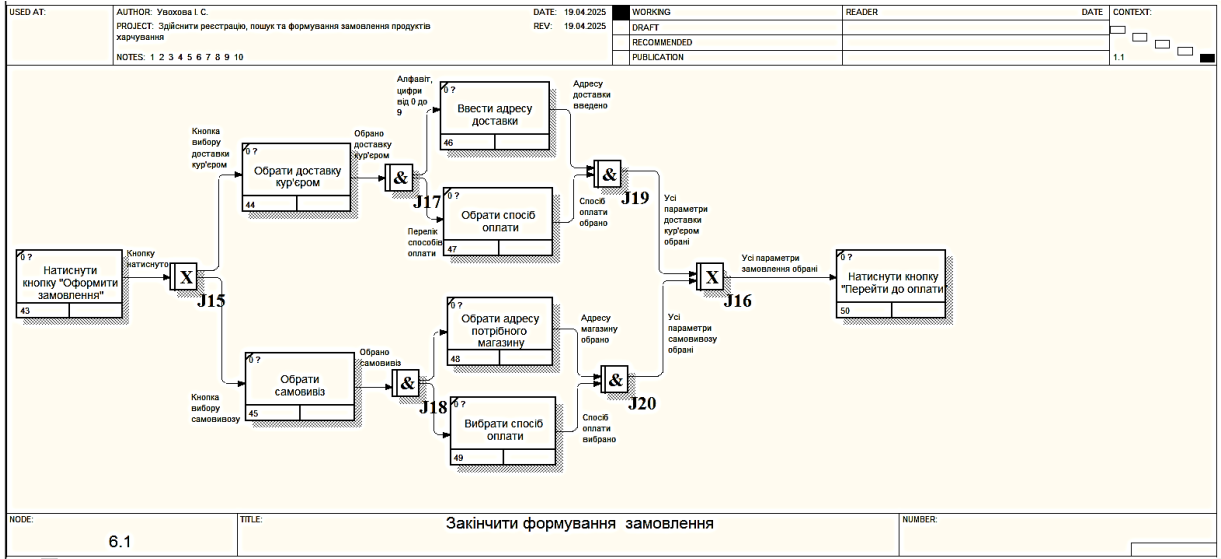


Рисунок 2.8 – Діаграма декомпозиції другого рівня процесу «Закінчити формування замовлення»

У результаті створення діаграм можна побачити п'ять моделей, в яких покроково описано п'ять основних процесів, з якими стикається користувач під час здійснення реєстрації, пошуку та формування замовлення продуктів харчування.

Наступним етапом є моделювання діаграми класів (рис. 2.9).

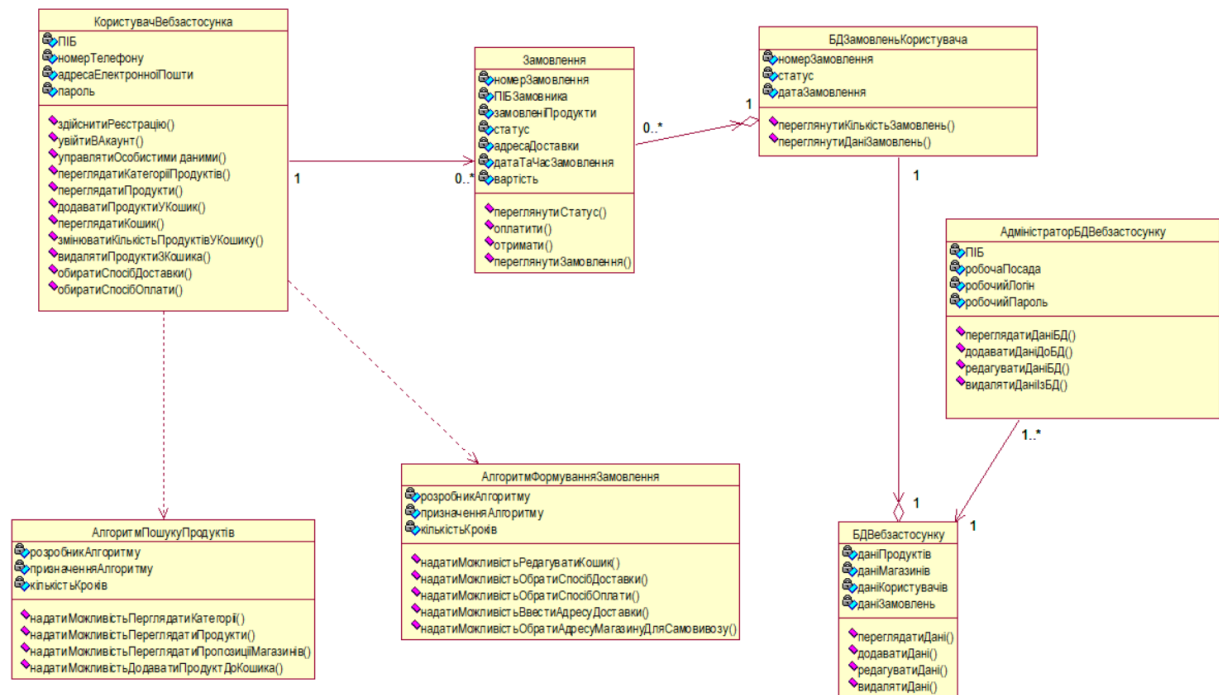


Рисунок 2.9 – Діаграма класів вебзастосунку для пошуку та формування замовлення продуктів харчування

У результаті моделювання, на діаграмі можна побачити 7 класів, які описують основну структуру системи:

- користувач вебзастосунку;
- замовлення;
- БД замовлень користувача;
- алгоритм пошуку продуктів;
- алгоритм формування замовлення;
- БД вебзастосунку;
- адміністратор БД вебзастосунку.

Останньою буде змодельовано діаграму прецедентів.

Опис об'єктів діаграми прецедентів вебзастосунку для пошуку та формування замовлення продуктів харчування:

- «Комплектувальник замовлень» – це людина, яка потребує переглянути список потрібних продуктів та здійснює збір замовлення;

– «Користувач» – це людина, яка потребує зареєструватися у вебзастосунку, увійти в особистий акаунт, обрати потрібні товари і сформулювати замовлення;

– «Кур'єр» – це людина, яка здійснює забір замовлення з магазину та його доставку користувачу;

– «Система сповіщення вебзастосунку» – це внутрішня система, що здійснює функцію сповіщення користувачів вебзастосунку щодо статусу та оплати замовлення;

– «Адміністратор БД вебзастосунку» – це людина, яка здійснює управління базою даних вебзастосунку, додає, редагує та видаляє дані;

– «Зареєструватися у вебзастосунку» – цей варіант використання ініціюється користувачем. Він забезпечує користувачу можливість зареєструватися у застосунку шляхом введення свого номеру телефону, адреси електронної пошти, паролю і своїх імені та прізвища;

– «Увійти в особистий акаунт» – цей варіант використання ініціюється користувачем. Він забезпечує користувачу можливість увійти у вже зареєстрований акаунт шляхом введення своєї адреси електронної пошти та паролю;

– «Переглянути товари» – цей варіант використання ініціюється користувачем. Він забезпечує користувачу можливість переглядати варіанти продуктів та шукати потрібні;

– «Додати товар до кошика» – цей варіант використання ініціюється користувачем. Він забезпечує користувачу можливість додавати потрібні користувачу продукти до кошика;

– «Відредагувати кошик» – цей варіант використання ініціюється користувачем. Він забезпечує користувачу можливість змінити кількість кожного з продуктів кошика або видалити непотрібні товари;

– «Оформити доставку замовлення» – цей варіант використання ініціюється користувачем. Він забезпечує користувачу можливість вказати усі необхідні дані для доставки замовлення;

– «Переглянути статус замовлення» – цей варіант використання ініціюється користувачем. Він забезпечує користувачу можливість переглянути на якому етапі доставки знаходиться його замовлення;

– «Переглянути склад замовлення» – цей варіант використання ініціюється комплектувальником замовлень. Він забезпечує комплектувальнику можливість переглянути список замовлених продуктів;

– «Зібрати замовлення» – цей варіант використання ініціюється комплектувальником замовлень. Він забезпечує комплектувальнику можливість зібрати потрібні продукти у торби задля подальшого відправлення;

– «Забрати замовлення з магазину» – цей варіант використання ініціюється кур'єром. Він забезпечує кур'єру можливість забрати зібране замовлення із точки видачі;

– «Доставити замовлення» – цей варіант використання ініціюється кур'єром. Він забезпечує кур'єру можливість доставити та віддати користувачу замовлення;

– «Сповістити користувача про успішну оплату» – цей варіант використання ініціюється системою сповіщення вебзастосунку. Він забезпечує системі можливість надіслати користувачу сповіщення про те, що його замовлення успішно сплачено;

– «Сповістити користувача про зміну статусу замовлення» – цей варіант використання ініціюється системою сповіщення вебзастосунку. Він забезпечує системі можливість надіслати користувачу сповіщення про те, що статус його замовлення змінено;

– «Додати дані у БД» – цей варіант використання ініціюється адміністратором БД вебзастосунку. Він забезпечує адміністратору можливість додавати інформацію до таблиць бази даних вебзастосунку;

– «Редагувати дані БД» – цей варіант використання ініціюється адміністратором БД вебзастосунку. Він забезпечує адміністратору можливість редагувати дані таблиць вебзастосунку;

– «Видалити дані з БД» – цей варіант використання ініціюється адміністратором БД вебзастосунку. Він забезпечує адміністратору можливість видалити непотрібні дані з таблиць вебзастосунку.

Діаграму прецедентів подано на рисунку 2.10.

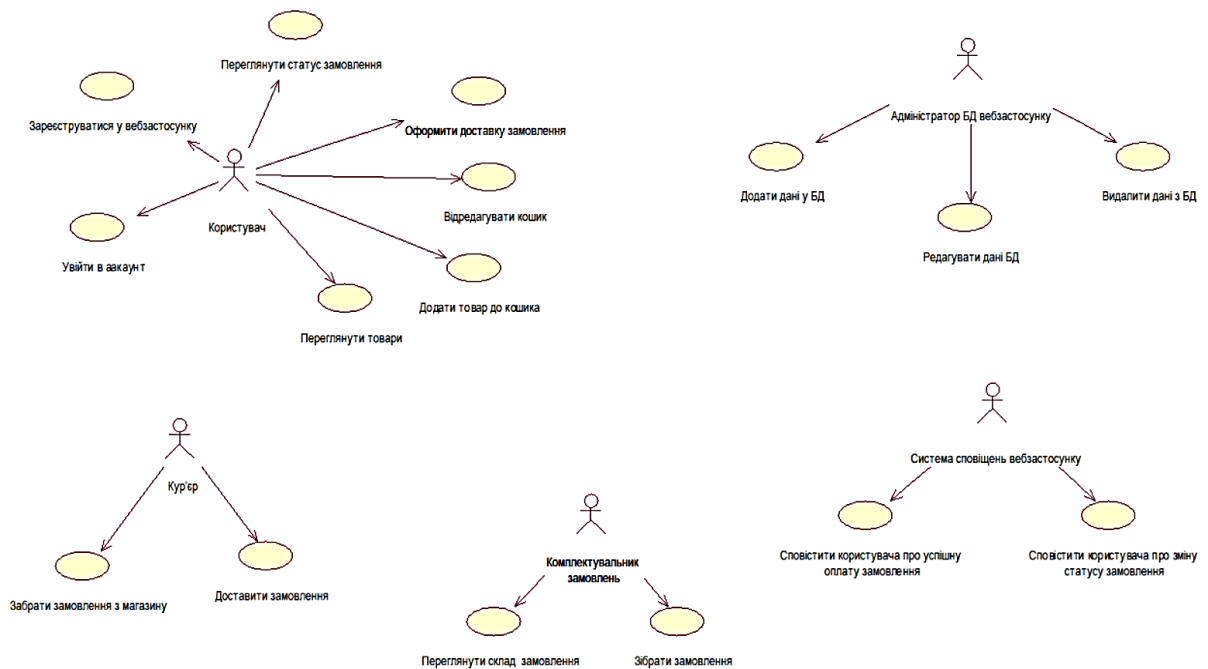


Рисунок 2.10 – Діаграма прецедентів вебзастосунку для пошуку та формування замовлення продуктів харчування

У результаті моделювання створено діаграму прецедентів, яка містить 5 акторів та 16 варіантів використання.

Розроблення структури системи є ключовим етапом її створення. Тож в цьому розділі було змодельовано основні діаграми вебзастосунку для пошуку та формування замовлення продуктів харчування.

## 3 РОЗРОБЛЕННЯ ВЕБЗАСТОСУНКУ ДЛЯ ПОШУКУ ТА ФОРМУВАННЯ ЗАМОВЛЕННЯ ПРОДУКТІВ ХАРЧУВАННЯ

### 3.1 Вибір інструментальних засобів для реалізації поставленої задачі

Ще одним важливим етапом розробки вебзастосунку є вибір інструментальних засобів для його реалізації. У сучасному світі ІТ існує безліч програмних середовищ, які б дозволяли працювати з кодом. Деякі з них знаходяться ще на етапі впровадження в життя розробників, а деякі вже стали невід'ємною частиною інструментарію сучасного програміста. Вибір інструментальних засобів залежить від багатьох факторів, таких як досвід розробника, функціональні вимоги до вебзастосунку, його типу, важливість інтеграції з іншими сервісами тощо. Для реалізації даного проекту було обрано такі засоби розробки, як Microsoft Visual Studio Code, мова JavaScript (бібліотека React і середовище виконання Node.js) та мова SQL.

Microsoft Visual Studio Code – програмне середовище, що використовується для розробки програмного забезпечення, написання та налагодження коду, а також тестування його роботи. Переваги VS Code:

- безкоштовність: програма є абсолютно безкоштовною, має відкритий код;
- легкість та швидкість: VS Code швидко встановлюється, не навантажує систему та працює стабільно навіть на слабших комп'ютерах;
- велика кількість розширень: середовище має тисячі додаткових розширень, переважна більшість з яких є безкоштовною;
- підтримка великої кількості мов програмування: редактор без додаткових розширень підтримує близько 30 мов програмування, що дозволяє писати як frontend, так і backend в одному середовищі;
- автодоповнення та підказки: за допомогою вбудованого механізму, показує автодоповнення коду та робить перевірку на помилки в реальному часі;

- підтримка форматування коду: VS Code, завдяки розширенням, допомагає тримати код «чистим» та читабельним;
- вбудований відлагоджувач: можна запускати код одразу в редакторі, ставити точки «стопу» та відслідковувати зміни.

JavaScript – динамічна мова програмування що, в контексті даного проєкту, дала можливість ефективно поєднати React (для створення інтерфейсу користувача) з Node.js (для створення серверної частини вебзастосунку). Переваги мови JavaScript:

- швидкість: код виконується напряму в браузері користувача, що дає змогу будувати інтерактивні інтерфейси без затримок з боку серверної частини;
- універсальність: працює як на клієнтській стороні, так і на серверній, що дозволяє створювати повноцінні вебзастосунки однією мовою;
- підтримка в усіх браузерах: мова підтримується усіма сучасними браузерами, без встановлення додаткових плагінів, що робить застосунок більш доступним;
- велика екосистема: JavaScript має багато бібліотек та фреймворків, які пришвидшують роботу;
- постійний розвиток: мова активно розвивається, за рахунок чого постійно становиться більш зручною та ефективною.

React – розроблена компанією Meta бібліотека для створення користувацьких інтерфейсів, що дозволяє будувати інтерактивні вебзастосунки. Переваги бібліотеки React:

- компонентна структура: код розбивається на окремі багаторазові блоки, що значно полегшує його читання та розробку;
- висока продуктивність: React підтримує швидке рендерення;
- активні розвиток та підтримка: швидко розвивається та має велику спільноту;
- масштабованість: зручно створювати як невеликі, так і складні системи.

Node.js – серверне середовище виконання для JavaScript. Завдяки базуванню на движку V8 дозволяє ефективно функціонувати навіть при великій кількості користувачів одночасно.

SQL – мова структурованих запитів для роботи з реляційною базою даних. Вона підтримується величезною кількістю СУБД та програмних середовищ, дозволяє створювати, редагувати та видаляти таблиці у базі даних, «діставати» потрібну інформацію та застосовувати до неї фільтри.

### 3.2 Етапи розроблення вебзастосунку для пошуку та формування замовлення продуктів харчування

Розроблення вебзастосунку – це не швидкий процес, який, задля спрощення, зазвичай, розбивають на етапи. У даній роботі процес розроблення вебзастосунку для пошуку та формування замовлення продуктів харчування був поділений на вісім основних етапів.

Першим етапом був аналіз існуючих вебзастосунків такого типу, їх кількості та якості, що дало змогу виділити основні переваги та недоліки аналогів. Також цей етап допоміг зрозуміти реальну кількість конкурентно-спроможних платформ в Україні.

На другому етапі було описано аналіз літературних джерел, що відповідають тематиці розроблюваного вебзастосунку. Під час опрацювання статей та наукових робіт було вивчено актуальні технології, що можуть допомогти під час створення платформи, а також психологічні аспекти, що впливають на людську поведінку під час закупівлі продуктів.

Третім етапом є моделювання структури вебзастосунку. Цей етап можна вважати одним із найбільш об'ємних у даній роботі, адже побудова структури системи із різних сторін займає багато часу та потребує обширного та конкретизованого опису.

Четвертий етап – вибір засобів побудови та реалізація бази даних для розроблюваного застосунку. Цей етап також є об’ємним і одним із найважливіших, адже на основі змодельованої БД в майбутньому будується логіка функціональних компонентів.

П’ятим етапом є вибір інструментальних засобів для програмної реалізації. На цьому етапі розробник має можливість ознайомитися із великою кількістю існуючих середовищ, мов програмування тощо та обрати саме ті, які є найбільш функціонально-підходящими для нього.

Шостий етап – розробка клієнтської частини вебзастосунку. Створення її структури та планування функціональних можливостей.

Сьомий етап – розробка серверної сторони застосунку. Написання усіх функцій, налагодження зв’язків із БД, створення API-ендпоінтів для обміну даними з клієнтською стороною, реалізація задуманої логіки.

Останній восьмий етап присвячений тестуванню вебзастосунку, опису його сильних сторін та перспектив розробки. На цьому етапі аналізуються та порівнюються мета роботи та отриманий результат.

У цьому розділі будуть описані два етапи програмної реалізації, а саме: розроблення клієнтської та серверної частин застосунку.

### 3.2.1 Розроблення клієнтської частини вебзастосунку для пошуку та формування замовлення продуктів харчування

Створення frontend-у є творчим процесом, під час якого розробник має зрозуміти, як саме повинен виглядати готовий варіант вебзастосунку. Кнопки, посилання, меню, зображення, кольори, таблиці, анімаційні ефекти – усе це «малюється» саме тут. Також не варто забувати й те, що усі ці елементи мають бути не лише привабливими, а й максимально зручними та зрозумілими для користувача, тож ось деякі правила, які було прийнято до уваги під час створення клієнтської частини:

– дизайн інтерфейсу має бути виконаним в одному стилі, без використання занадто яскравих відтінків, аби не викликати у користувача реакцію відторгнення;

– елементи сторінок мають бути розташовані симетрично один одному, не виходячи за рамки екрану;

– усі елементи мають бути оптимального розміру (наприклад: розмір кнопок повинен бути відповідного, до її вмісту, розміру);

– шрифт тексту сторінок не має бути занадто малим, аби людина спокійно могла його прочитати, але і не повинен бути занадто великим;

– колір тексту має бути контрастним відповідно до фону (найкраще підійдуть темні кольори тексту для світлого фону та світлі – для темного фону);

– у вебзастосунку не має бути жодних неактивних кнопок, усі кнопки та посилання повинні бути клікабельними;

– при натисканні на поля вводу, вони стають активними, а в середині з'являється курсор;

– текст усіх кнопок має однозначно описувати дію, яку вони виконують.

У рамках виконання даного проєкту було створено дванадцять сторінок вебзастосунку:

– RegisterPage: сторінка реєстрації користувачів, містить поля вводу (рис. 3.1) та кнопку «Зареєструватися»;

```
<form onSubmit={handleSubmit}>
  <input
    name="name"
    placeholder="Ім'я"
    value={form.name}
    onChange={handleChange}
    required
  />
  {validationErrors.name && (
    <p className="error">{validationErrors.name}</p>
  )}
}
```

Рисунок 3.1 – Приклад реалізації виведення поля форми реєстрації

– LoginPage: сторінка входу в акаунт, містить поля вводу, кнопки «Увійти» та «Зареєструватися», також реалізовані спливаючі сповіщення про успішний вхід (рис. 3.2) або помилку;

```
const handleLogin = async (e) => {
  e.preventDefault();
  try {
    const res = await API.post("/users/login", form);
    localStorage.setItem("token", res.data.token);
    localStorage.setItem("userId", res.data.userId);
    toast.success("Вхід успішний!");
  }
}
```

Рисунок 3.2 – Приклад реалізації виведення сповіщення на сторінці

– HomePage: головна сторінка вебзастосунку, містить перелік категорій товарів, перелік доступних магазинів та поле пошуку продуктів (рис. 3.3);

```
useEffect(() => {
  if (search.trim()) {
    API.get(`/products/search?q=${search}`).then(res => setSearchResults(res.data));
  } else {
    setSearchResults([]);
  }
}, [search]);
```

Рисунок 3.3 – Приклад реалізації виведення пошукового поля на головній сторінці

– ProductPage: сторінка кожного продукту, відображає інформацію про товар, а також про вартість в різних магазинах, містить кнопки «Додати до кошику» (рис. 3.4) та «Перейти в магазин»;

```
const quantity = quantities[offeringId] || 1;
API.post("/cart/add", {
  userId,
  storeId,
  offeringId,
  quantity,
  price,
}).then((res) => {
  toast.success("Товар додано до кошика!", { ...
});
});
```

Рисунок 3.4 – Приклад виклику функції додавання товару до кошику

– CategoryProductsPage: сторінка категорій товарів, містить назву категорії та перелік продуктів, що до неї входять (рис. 3.5);

```

<div
  key={p.Id_Product}
  className="product-card-categoryPage"
  onClick={() => navigate(`/product/${p.Id_Product}`)}
>
  {p.Product_Image && (
    <img
      src={`data:image/png;base64,${p.Product_Image}`}
      alt={p.Product_Name}
      className="product-image-categoryPage"
    />
  )}
}

```

Рисунок 3.5 – Реалізація виведення товарів у категорії

– StorePage: сторінка магазину: містить інформацію про магазин, перелік категорій (рис. 3.6) і товарів, що в ньому є;

```

useEffect(() => {
  if (viewMode === "categories") {
    API.get(`/stores/${id}/categories`)
      .then((res) => setCategories(res.data))
      .catch((err) => console.error("Помилка при завантаженні категорій:", err));
  } else if (viewMode === "products") {

```

Рисунок 3.6 – Реалізація виведення категорій, що є у магазині

– CartPage: кошик, містить у собі перелік кошиків магазинів з інформацією про додані продукти (рис. 3.7);

```

const updateQuantity = (itemId, newQty) => {
  if (newQty <= 0) {
    API.delete(`/cart/item/${itemId}`).then((res) => {
      reloadCarts();
    });
  } else {
    API.put(`/cart/item/${itemId}`, { quantity: newQty }).then(() => {
      reloadCarts();
    });
  }
};

```

Рисунок 3.7 – Реалізація виклику функції зміни кількості товару у кошику

– CheckoutPage: сторінка оформлення замовлення, містить поля вводу та кнопку «Підтвердити замовлення» (рис. 3.8);

```
const handleSubmit = () => {
  if (!validateForm()) return;

  API.post("/orders/place", {
    userId,
    storeId,
    totalPrice,
    deliveryTypeId: form.deliveryTypeId,
    deliveryAddress: form.address,
    paymentType: form.paymentType,
    status: "В процесі",
    items: cartItems,
  })
  .then((response) => {
    toast.success("Замовлення успішно оформлено!", {
      position: "top-right",
      autoClose: 1500,
      hideProgressBar: true,
      closeButton: false,
    });
  });
}
```

Рисунок 3.8 – Реалізація виклику функції надсилання форми

– ResultOrderPage: сторінка інформації по тільки-но оформленому замовленню (рис. 3.9);

```
useEffect(() => {
  if (orderId) {
    API.get(`/orders/one/${orderId}`)
      .then((res) => setOrder(res.data))
      .catch((err) => console.error("Помилка при завантаженні замовлення", err));
  }
}, [orderId]);

if (!order) {
  return <div className="order-loading">Завантаження...</div>;
}
```

Рисунок 3.9 – Реалізація виклику функції, що виводить деталі замовлення

– MyOrdersPage: сторінка замовлень, відображає перелік усіх оформлених користувачем покупок з основною інформацією (рис. 3.10);

```
const formatDate = (dateString) => {
  try {
    const date = new Date(dateString);
    return format(date, 'dd MMMM yyyy, HH:mm', { locale: uk });
  } catch {
    return dateString;
  }
};
```

Рисунок 3.10 – Приклад перетворення дати у потрібний формат

– ProfileEditPage: сторінка для редагування даних користувача, містить форму з полями (рис. 3.11) та кнопку «Оновити»;

```
if (/[a-zA-Zа-яА-ЯіїІІєєЄєҐґ]/.test(userData.phone)) {
  errors.phone = "Номер телефону не повинен містити літери";
} else if (!/^+38\d{10}$/.test(userData.phone)) {
  errors.phone = "Номер телефону має починатися з +38 та містити рівно 13 символів.";
}
```

Рисунок 3.11 – Приклад реалізації валідації даних поля «Номер телефону»

– ListPage: сторінка, на якій розміщено текстове поле для складання списку покупок (рис. 3.12).

```
useEffect(() => {
  if (userId) {
    const savedNote = localStorage.getItem(`note_${userId}`);
    if (savedNote) {
      setNote(savedNote);
    }
  }
}, [userId]);

const handleChange = (event) => {
  const newNote = event.target.value;
  setNote(newNote);

  if (userId) {
    localStorage.setItem(`note_${userId}`, newNote);
  }
};
```

Рисунок 3.12 – Приклад реалізації збереження списку продуктів у локальному сховищі

Також на всіх сторінках, окрім входу та реєстрації, було реалізовано шість кнопок: «Головна», «Кошик», «Мої замовлення», «Список товарів», «Профіль», та «Вийти» (рис. 3.13) для навігації по вебзастосунку.

```
const handleNavigation = (path) => {  
  if (!userId) {  
    navigate("/");  
  } else {  
    navigate(path);  
  }  
};
```

Рисунок 3.13 – Реалізація сценарію перенесення неавторизованого користувача на сторінку входу

### 3.2.2 Розроблення серверної частини вебзастосунку для пошуку та формування замовлення продуктів харчування

Написання backend-у, в свою чергу, є доволі складним та потребує уважності процесом, адже саме під час нього реалізується абсолютно увесь функціонал застосунку. Організація роботи з базою даних, забезпечення стабільності та коректної обробки усіх запитів з клієнтської частини – кожна деталь впливає на якість та ефективність роботи цілої системи.

В ході розробки серверної частини вебзастосунку для пошуку та формування замовлення продуктів харчування було підключено, заздалегідь розроблену, БД SQLite та реалізовано програмний код для таких функцій:

- реєстрація: описано логіку занесення даних користувача у БД, додано перевірку на однакові електронні адреси та хешування паролю (рис. 3.14);

- вхід в акаунт: реалізовано вивід форми із полями та перевірку наявності користувача в БД (рис. 3.15);

```
// Реєстрація користувача
router.post("/register", async (req, res) => {
  const { name, surname, email, phone, password } = req.body;

  // Перевірка, чи вже існує користувач із таким email
  db.get(`SELECT * FROM Users WHERE User_Email = ?`, [email], async (err, existingUser) => {
    if (err) return res.status(500).json({ error: "Помилка при перевірці користувача" });
    if (existingUser) return res.status(400).json({ error: "Користувач з такою адресою вже існує" });

    // Якщо email унікальний, продовжуємо реєстрацію
    try {
      const hashed = await bcrypt.hash(password, 10);
      const sql = `INSERT INTO Users (User_Name, User_Surname, User_Email, Phone, User_Password) VALUES (?, ?, ?, ?, ?)`;

      db.run(sql, [name, surname, email, phone, hashed], function (err) {
        if (err) return res.status(500).json({ error: err.message });
        res.status(201).json({ userId: this.lastID });
      });
    } catch (hashErr) {
      res.status(500).json({ error: "Помилка при обробці пароля" });
    }
  });
});
```

Рисунок 3.14 – Реалізація функції реєстрації користувача

```
router.post("/login", (req, res) => {
  const { email, password } = req.body;
  db.get(`SELECT * FROM Users WHERE User_Email = ?`, [email], async (err, user) => {
    if (err || !user) return res.status(404).json({ error: "Користувача не знайдено" });
    const match = await bcrypt.compare(password, user.User_Password);
    if (!match) return res.status(401).json({ error: "Невірний пароль" });
    const token = jwt.sign({ id: user.Id_User }, "SECRET", { expiresIn: "2h" });
    res.json({ token, userId: user.Id_User });
  });
});
```

Рисунок 3.15 – Реалізація функції входу користувача

– пошук товарів за назвою: товари шукаються незалежно від регістру та частини введеного слова (рис. 3.16);

```
router.get("/search", (req, res) => {
  const q = req.query.q || '';
  const qStr = `%${q.toLocaleLowerCase()}%`;
  db.all("SELECT * FROM Products WHERE Product_Name_Lower LIKE ?", [qStr], (err, rows) => {
    if (err) return res.status(500).json({ error: err.message });
    const withImages = rows.map(p => ({
      ...p,
      Product_Image: p.Product_Image ? Buffer.from(p.Product_Image).toString("base64") : null
    }));
    res.json(withImages);
  });
});
```

Рисунок 3.16 – Реалізація функції пошуку товару за назвою

– виведення продуктів, магазинів та категорій на сторінку: усі дані витягуються із БД та розташовуються в залежності від типу сортування (рис. 3.17);

```
router.get("/categories", (req, res) => {
  db.all("SELECT * FROM Categories ORDER BY Category_Name COLLATE NOCASE ASC", [], (err, rows) => {
    if (err) return res.status(500).json({ error: err.message });
    res.json(rows);
  });
});
```

Рисунок 3.17 – Реалізація виведення усіх категорій в алфавітному порядку

– дії з кошиком: реалізовано занесення, редагування кількості та видалення продуктів з кошику та БД, а також підрахунок їх загальної вартості (рис. 3.18);

```
router.put("/item/:id", (req, res) => {
  const { id } = req.params;
  const { quantity } = req.body;

  if (quantity === 0) {
    db.get(`SELECT Cart_Id FROM Carts_Items WHERE Id_Cart_Items = ?`, [id], (err, item) => {
      if (err) return res.status(500).json({ error: err.message });
      const cartId = item.Cart_Id;

      db.run(`DELETE FROM Carts_Items WHERE Id_Cart_Items = ?`, [id], function (err) {
        if (err) return res.status(500).json({ error: err.message });
        updateCartTotal(cartId, res, { itemDeleted: true });
      });
    });
  } else {
    db.get(`SELECT Cart_Id FROM Carts_Items WHERE Id_Cart_Items = ?`, [id], (err, item) => {
      if (err) return res.status(500).json({ error: err.message });
      const cartId = item.Cart_Id;

      db.run(`UPDATE Carts_Items SET Quantity_Cart = ? WHERE Id_Cart_Items = ?`, [quantity, id], function (err) {
        if (err) return res.status(500).json({ error: err.message });
        updateCartTotal(cartId, res, { itemDeleted: false });
      });
    });
  }
});
```

Рисунок 3.18 – Реалізація зміни кількості товарів у кошику

– оформлення замовлення: описано логіку заповнення полів даних замовлення, вибір типів доставки та оплати, занесення усіх даних у таблиці БД (рис. 3.19);

```

db.run(`INSERT INTO Orders (User_Id_Order, Store_Id_Order, Date_of_Order,
    Total_Price_Order, Delivery_Type_Id, Delivery_Address, Type_of_Payment, Status)
    VALUES (?, ?, ?, ?, ?, ?, ?, ?)`,
[userId, storeId, date, totalPrice, deliveryTypeId, deliveryAddress, paymentType, status],
function (err) {
    if (err) return res.status(500).json({ error: err.message });

    const orderId = this.lastID;

    const insertItem = db.prepare(`INSERT INTO Order_Items
    (Order_Id, Offering_Id_Order, Quantity_Order, Price_for_Unit_Order)
    VALUES (?, ?, ?, ?)`);

    cartItems.forEach(item => {
        insertItem.run(orderId, item.Offering_Id_Cart, item.Quantity_Cart, item.Price_for_Unit_Cart);
    });
}

```

Рисунок 3.19 – Реалізація занесення даних замовлення

– редагування профілю: описано логіку виведення форми з поточними даними, які можливо змінити, працює лише якщо користувач зареєстрований (рис. 3.20).

```

// Оновлення даних користувача
router.put("/update/:userId", async (req, res) => {
    const { name, surname, email, phone, currentPassword, newPassword } = req.body;
    const userId = req.params.userId;

    // Функція зміни особистих даних
    const changeUserData = (name, surname, email, phone, userId) => {
        return new Promise((resolve, reject) => {
            const sql = `
                UPDATE Users SET
                User_Name = ?,
                User_Surname = ?,
                User_Email = ?,
                Phone = ?
                WHERE Id_User = ?
            `;

            db.run(sql, [name, surname, email, phone, userId], function (err) {
                if (err) return reject({ error: err.message });
                if (this.changes === 0) {
                    return reject({ error: "Користувача не знайдено" });
                }
                resolve({ message: "Дані успішно оновлені" });
            });
        });
    };
}

```

Рисунок 3.20 – Реалізація зміни даних користувача

Загалом на backend-і було створено 5 файлів, що відповідають за маршрутизацію запитів: userRoutes, productRoutes, storeRoutes, cartRoutes і orderRoutes.

### 3.3 Тестування розробленого вебзастосунку та аналіз результатів

Тестування вебзастосунку є останнім етапом його розроблення. Саме під час перевірки можна впевнитися, що інтерфейс працює без затримок, серверна частина обробляє інформацію коректно і усі заплановані функції виконуються вірно. У цьому підрозділі буде розглянуто тестування вебзастосунку для пошуку та формування замовлення продуктів харчування.

Для перевірки застосунку, його буде запуснено на локальному хості `http://localhost:3000`. Запустивши серверну та клієнтську частину, можна побачити, що відкривається перша сторінка вебзастосунку – сторінка входу (рис. 3.21).

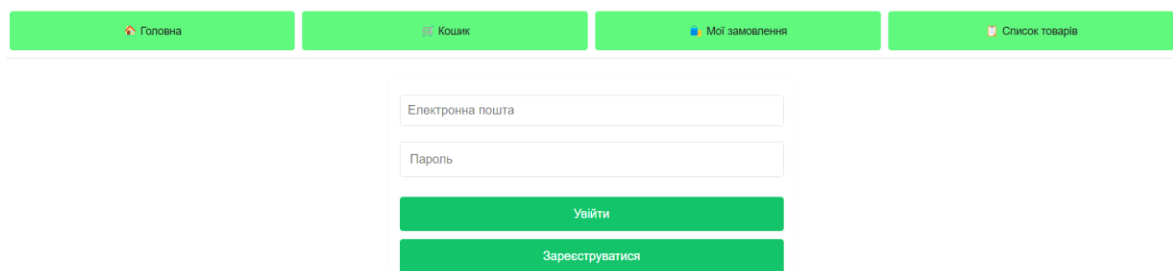


Рисунок 3.21 – Сторінка входу вебзастосунку для пошуку та формування замовлення продуктів харчування

Задля подальшої роботи, буде зареєстровано нового користувача, перейшовши на сторінку реєстрації. Для тестування валідації, на першому етапі буде введено некоректні дані, а на другому – коректні (рис. 3.22, рис. 3.23).

Також для перевірки функції, яка не допускає реєстрацію з однаковими електронними адресами, буде виконано спробу зареєструватися із вже існуючою у БД поштою (рис. 3.24).

Головна Кошик Мої замовлення Список товарів Увійти

Ім'я має містити щонайменше 2 символи.

Прізвище не повинно містити цифри.

Електронна пошта повинна містити символи '@' і '.'

Номер телефону не повинен містити літери.

Пароль має містити щонайменше 8 символів.

Рисунок 3.22 – Сторінка реєстрації вебзастосунку для пошуку та формування замовлення продуктів харчування, введено некоректні дані

Головна Кошик Мої замовлення Список товарів Увійти

Рисунок 3.23 – Сторінка реєстрації вебзастосунку для пошуку та формування замовлення продуктів харчування, введено коректні дані

Головна Кошик ❗ Користуван з такою електронною адресою вже існує. Список товарів Увійти

Рисунок 3.24 – Сторінка реєстрації вебзастосунку для пошуку та формування замовлення продуктів харчування, введено вже існуючу поштову адресу

Далі було введено дані тільки-но зареєстрованого користувача (рис. 3.25) і натиснуто кнопку «Увійти».

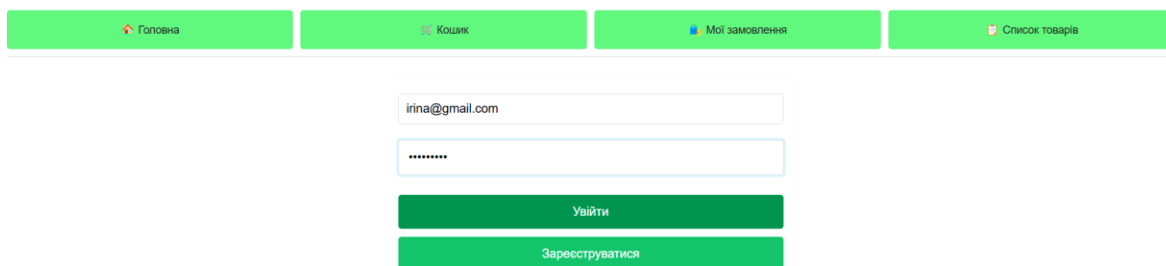


Рисунок 3.25 – Сторінка входу із заповненими даними вебзастосунку для пошуку та формування замовлення продуктів харчування

Увійшовши у систему, відкривається головна сторінка (рис. 3.26), на якій можна побачити перелік категорій та магазинів, а також рядок пошуку. Далі буде протестовано функції пошуку товару за назвою (рис. 3.27), переходу на сторінку категорії (рис. 3.28) та магазину (рис. 3.29), для перегляду доступних у них товарів.

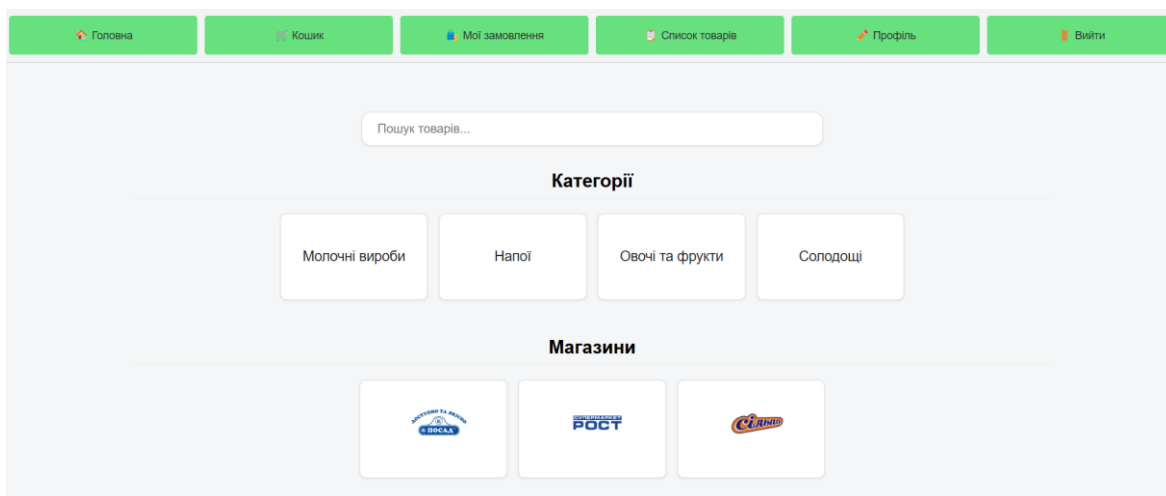


Рисунок 3.26 – Головна сторінка вебзастосунку для пошуку та формування замовлення продуктів харчування

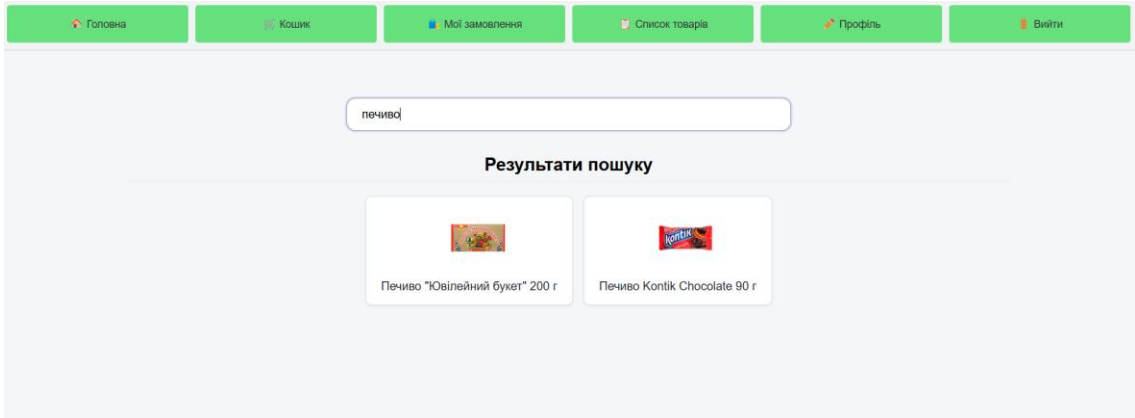


Рисунок 3.27 – Пошук товару у вебзастосунку для пошуку та формування замовлення продуктів харчування

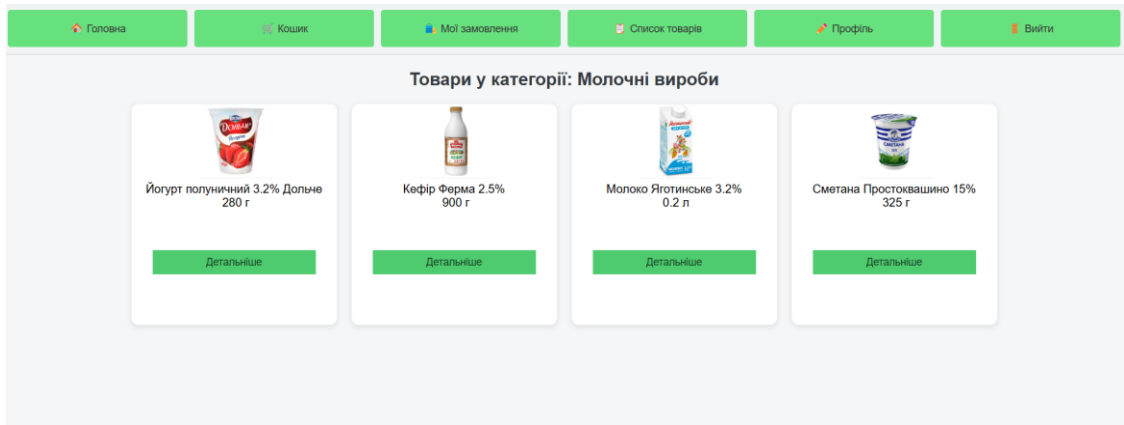


Рисунок 3.28 – Перегляд категорії товарів у вебзастосунку для пошуку та формування замовлення продуктів харчування

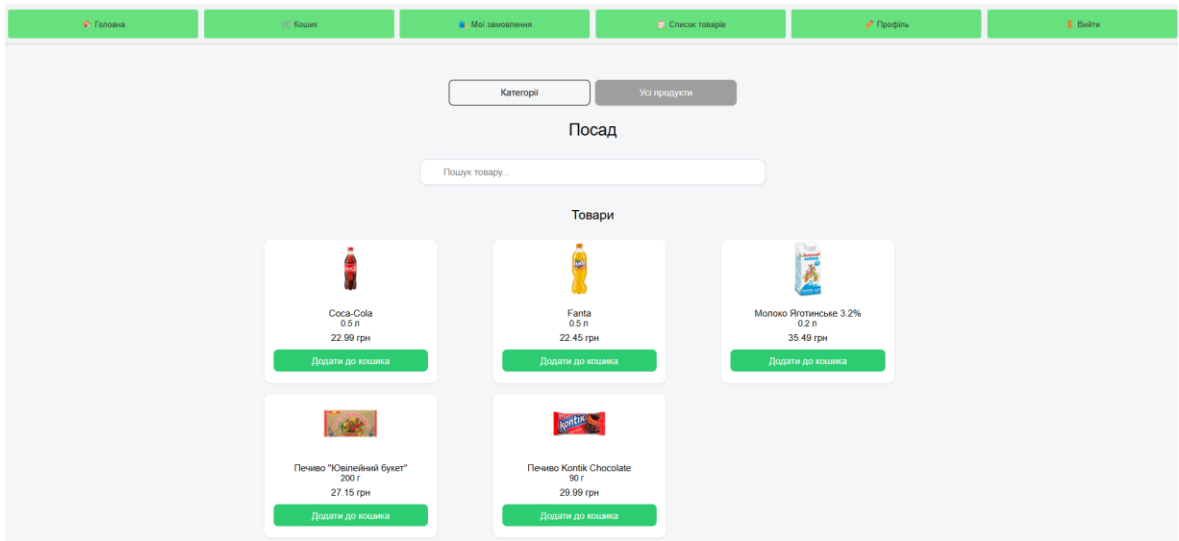


Рисунок 3.29 – Сторінка магазину вебзастосунку для пошуку та формування замовлення продуктів харчування

Також протестовано і функцію переходу на сторінку конкретного товару (рис. 3.30).

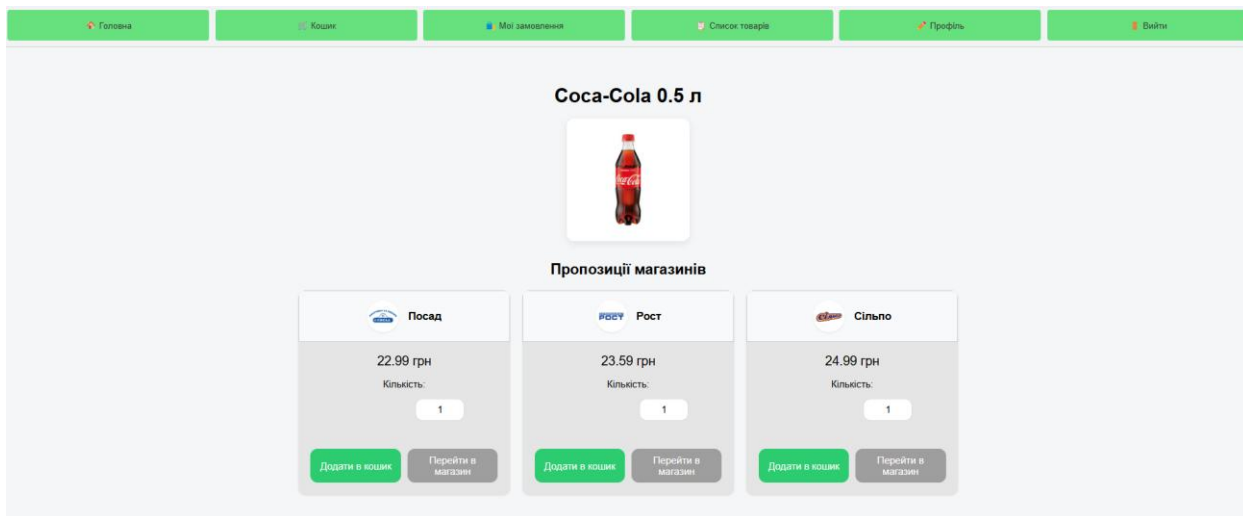


Рисунок 3.30 – Сторінка товару вебзастосунку для пошуку та формування замовлення продуктів харчування

Наступним кроком буде перевірка функції додавання товарів у кошик (рис. 3.31) та зміни кількості товару (рис. 3.32).

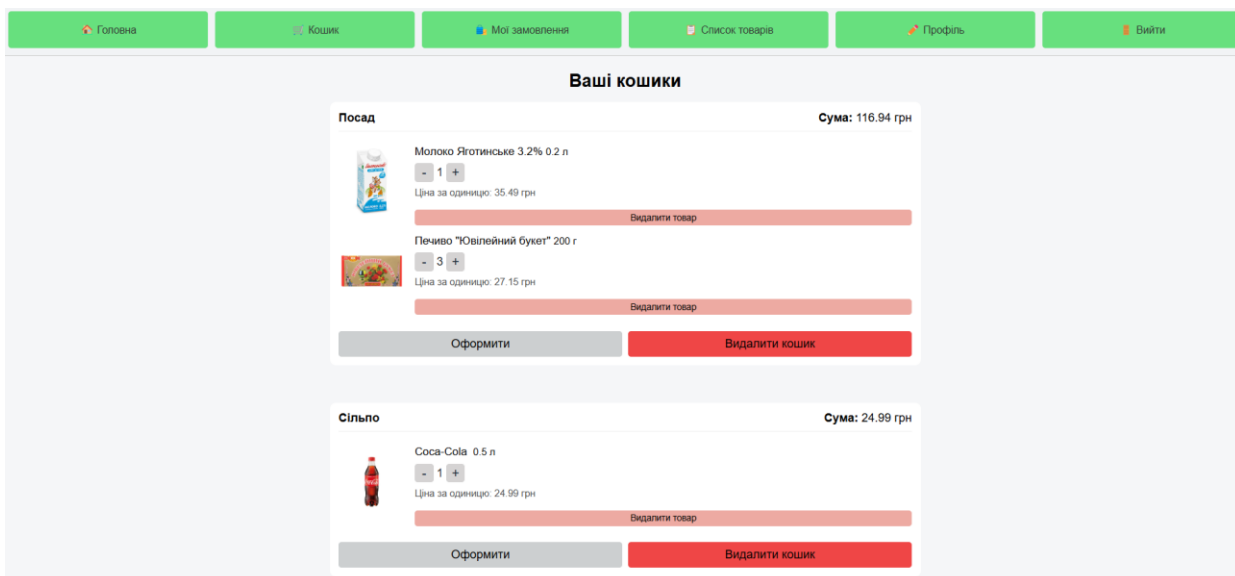


Рисунок 3.31 – Сторінка кошику вебзастосунку для пошуку та формування замовлення продуктів харчування

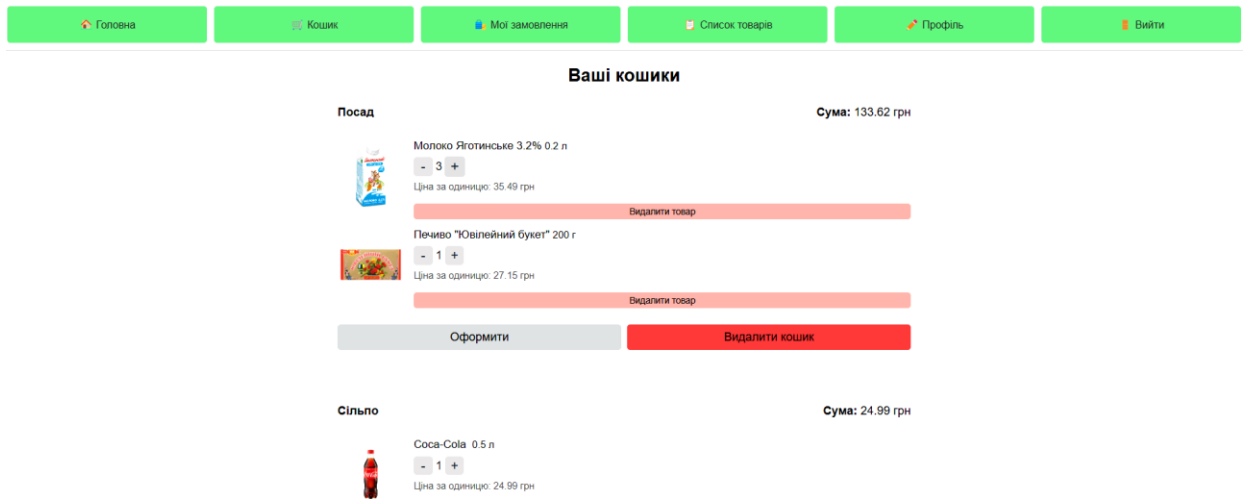


Рисунок 3.32 – Сторінка кошику зі зміненою кількістю товару вебзастосунку для пошуку та формування замовлення продуктів харчування

Подальшим кроком буде тестування функції оформлення замовлення (рис. 3.33 – рис. 3.36). Таким чином у цьому пункті буде перевірено валідацію полів, виведення сторінки результату оформлення і додавання замовлення на сторінку «Мої замовлення».

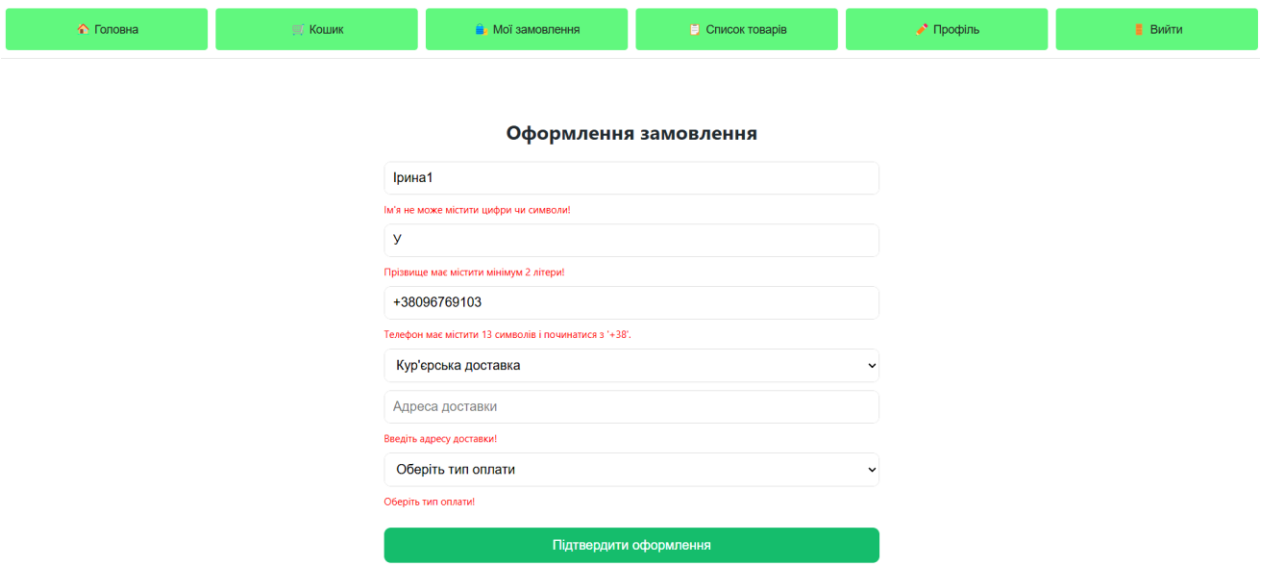


Рисунок 3.33 – Сторінка оформлення замовлення вебзастосунку для пошуку та формування замовлення продуктів харчування (у форму введено некоректні дані)

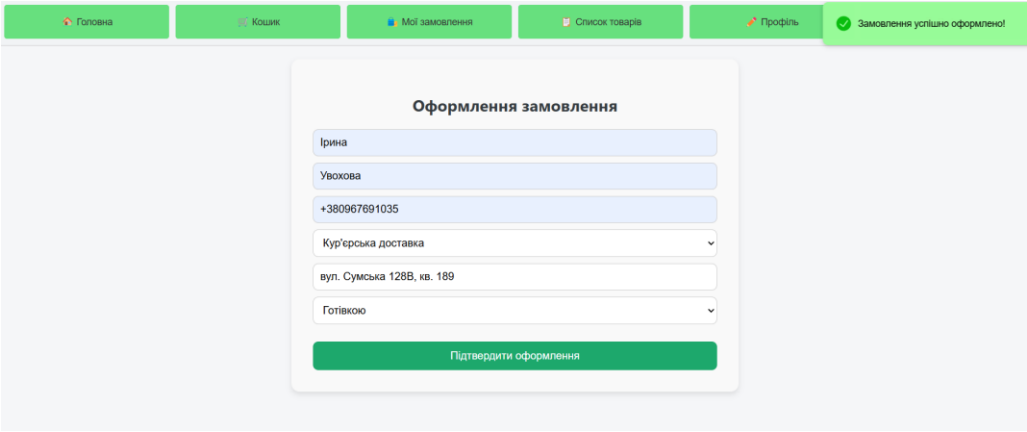


Рисунок 3.34 – Сторінка оформлення замовлення вебзастосунку для пошуку та формування замовлення продуктів харчування (у форму введено коректні дані)

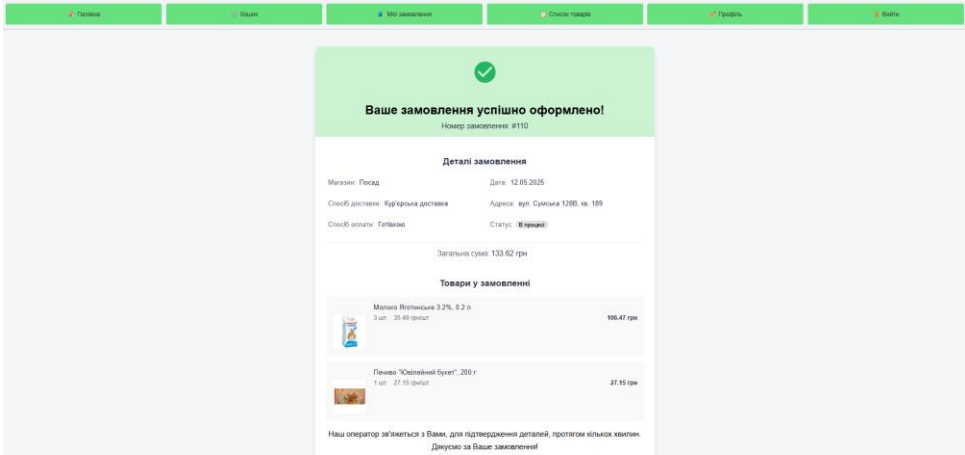


Рисунок 3.35 – Сторінка оформленого замовлення вебзастосунку для пошуку та формування замовлення продуктів харчування

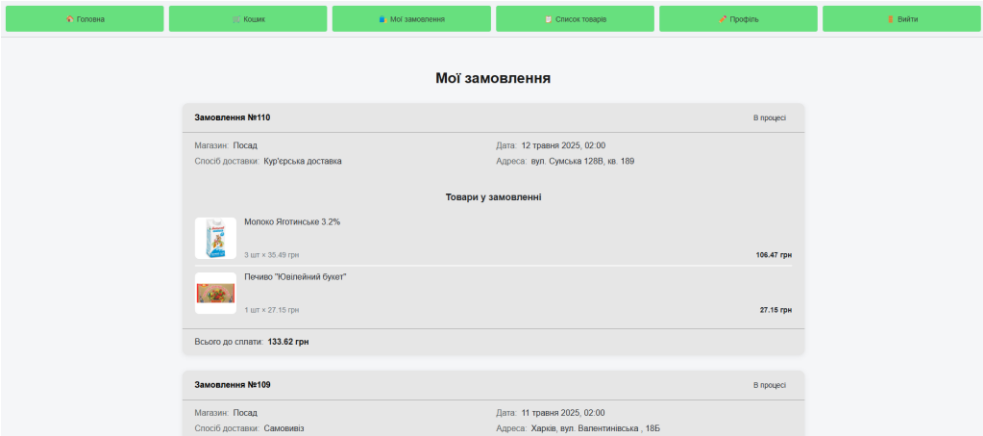
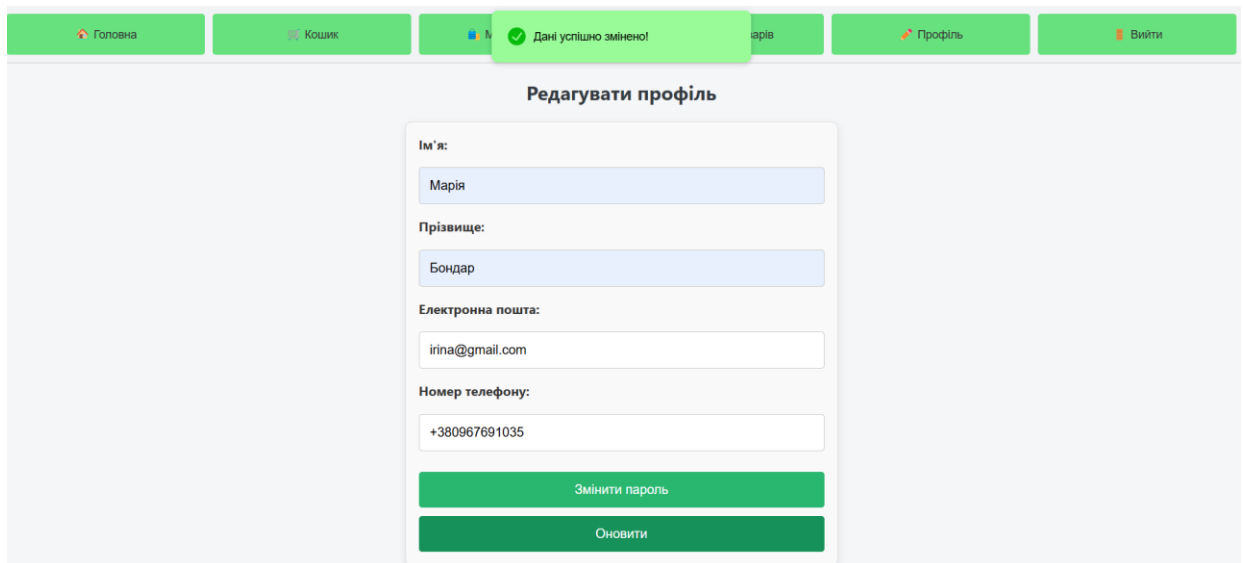


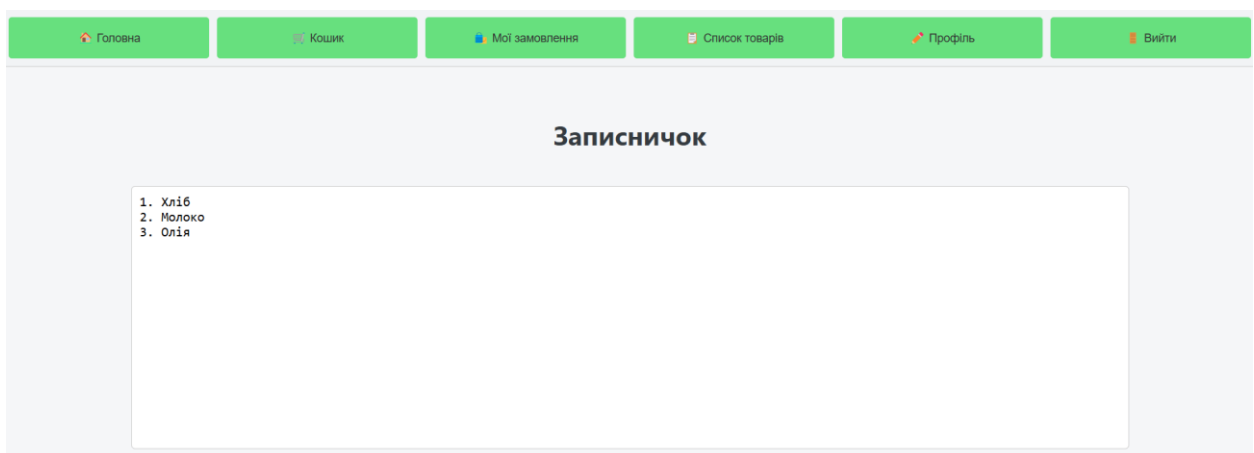
Рисунок 3.36 – Сторінка замовлень вебзастосунку для пошуку та формування замовлення продуктів харчування

Наостанок також буде протестовано функції зміни даних користувача (рис. 3.37) та написання списку товарів (рис. 3.38).



The screenshot shows a web application interface with a green navigation bar at the top containing links for 'Головна', 'Кошик', 'Мі', 'Список товарів', 'Профіль', and 'Вийти'. A notification banner above the main content area reads 'Дані успішно змінено!'. The main content area is titled 'Редагувати профіль' and contains a form with the following fields: 'Ім'я:' (filled with 'Марія'), 'Прізвище:' (filled with 'Бондар'), 'Електронна пошта:' (filled with 'irina@gmail.com'), and 'Номер телефону:' (filled with '+380967691035'). At the bottom of the form are two buttons: 'Змінити пароль' and 'Оновити'.

Рисунок 3.37 – Сторінка зміни даних користувача вебзастосунку для пошуку та формування замовлення продуктів харчування



The screenshot shows a web application interface with a green navigation bar at the top containing links for 'Головна', 'Кошик', 'Мої замовлення', 'Список товарів', 'Профіль', and 'Вийти'. The main content area is titled 'Записничок' and contains a list of items: '1. Хліб', '2. Молоко', and '3. Олія'.

Рисунок 3.38 – Сторінка із записничком вебзастосунку для пошуку та формування замовлення продуктів харчування

В кінці тестування також перевірено таблиці бази даних, до яких коректно було занесено інформацію про зареєстрованого користувача (рис. 3.39) та оформлене замовлення (рис. 3.40).

	Id_User	User_Name	User_Surn	User_Email	Phone	User_Password
1	44	Ірина	Увохова	irina@gmail.com	+380961761864	\$2b\$10\$a1e47piGzbus363GhJB90eQ7SrvMohRo7gCXdw3zzB/Y572o5OCle

Рисунок 3.39 – Таблиця Users БД вебзастосунок для пошуку та формування замовлення продуктів харчування

	Id_Order	User_Id_Order	Store_Id_Order	Date_of_Order	Total_Price_Order	Delivery_Type_Id	Delivery_Address	Status	Type_of_D
1	106	44	3	2025-05-10	88.47	2	Харків, вул. Валентинівська, 185	В процесі	Готівкою
2	107	44	3	2025-05-11	116.94	1	Ак. Павлова 134/16, 189	В процесі	Готівкою

Рисунок 3.40 – Таблиця Orders БД вебзастосунок для пошуку та формування замовлення продуктів харчування

Отже, протестувавши вебзастосунок, можна сказати, що запланований функціонал працює повністю правильно, backend коректно комунікує з БД, а інтерфейс користувача є доволі спокійним та приємним для очей.

### 3.4 Перспективи подальшої роботи

У результаті розробки вебзастосунок для пошуку та формування замовлення продуктів харчування було виконано усі, поставлені в рамках цієї роботи, задачі та реалізовано основний функціонал даної системи. Отже, можна сказати, що застосунок готовий до використання, проте йому ще є, куди розвиватися, адже розробити складний застосунок з широким та різноманітним функціоналом у рамках одного проекту майже неможливо.

Із найбільш ймовірних покращень даного вебзастосунок можна виділити наступні:

– інтеграція з Google Maps: додання карти, задля легшого знаходження розташування магазинів та побудова маршруту до них є одним із найбільших кроків до поліпшення функціоналу;

– додавання функції оплати карткою: оплата онлайн стала невід’ємною частиною нашого життя, тож впровадження даної опції у застосунок буде гарним рішенням;

– автоматизація процесу внесення даних про продукти: це дозволить отримувати найбільш актуальну інформацію про ціни та наявність товару безпосередньо від магазинів;

– покращення безпеки під час процесу обробки замовлень: використання безпекових протоколів та зчитування інформації напряму зі сторінки, а не з БД дозволить максимально посилити безпеку даних користувачів;

– додавання функції зчитування QR-коду: ця функція значно пришвидшить пошук товарів, що спровокує більшу зацікавленість серед потенційних користувачів;

– додавання функції відстежування замовлення: урізноманітнити варіанти статусу, аби користувач міг зрозуміти, на якій стадії знаходиться його замовлення (наприклад додати статуси «Опрацьовується», «Збирається», «Очікує кур’єра», «В дорозі» та «Завершено»);

– розширення списку категорій товарів: окрім продуктів харчування, надавати інформацію і про інші товари;

– додавання більшої кількості фільтрів для магазинів: сортування по відстані, показ магазинів у конкретному радіусі, фільтрація по кільком полям – усе це допоможе користувачам знайти найбільш оптимальний варіант для себе.

## ВИСНОВКИ

Як показав аналіз останніх науково-технічних досягнень в галузі інформаційних технологій, на сьогоднішній день зазначена галузь має багато напрямків, які успішно вирішують багато класів прикладних задач [19–33].

У рамках даної кваліфікаційної роботи був розроблений вебзастосунок для пошуку та формування замовлення продуктів харчування. Для досягнення мети було виконано такі задачі:

- проаналізовано сучасні застосунки даного типу в Україні та закордоном;

- проаналізовано літературні джерела щодо розробки схожих вебзастосунків, програмних методів реалізації та впливу даних систем на магазини та психологічний стан користувачів;

- побудовано базу даних та змодельовано її ER-діаграму;

- змодельовано структуру вебзастосунку за допомогою діаграм класів, прецедентів та діаграми в нотації IDEF3;

- обрано та обґрунтовано інструментальні засоби для розробки;

- розроблено клієнтську частину вебзастосунку за допомогою бібліотеки React;

- розроблено серверну частину системи з використанням середовища виконання Node.js та мови програмування JavaScript;

- протестовано застосунок, методом запуску в локальному середовищі;

- описано перспективи подальшого розвитку.

Розроблений застосунок дозволяє користувачу порівнювати ціни на продукти харчування в різних магазинах не перемикаючись між їх сайтами, формувати кошик та оформлювати замовлення онлайн.

Результати роботи апробовано у вигляді тез доповіді під час Міжнародного молодіжного форуму «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ У ХХІ СТОЛІТТІ» [34].

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Tabletki.ua. URL: <https://tabletki.ua/> (дата звернення 08.04.2025).
2. Ceneo.pl. URL: <https://www.ceneo.pl> (дата звернення 08.04.2025).
3. Trolley.co.uk. URL: <https://www.trolley.co.uk/> (дата звернення 08.04.2025).
4. Trustpilot – Trolley.co.uk. URL: <https://uk.trustpilot.com/review/trolley.co.uk> (дата звернення 10.04.2025).
5. PN.com.ua. URL: <https://pn.com.ua/> (дата звернення 10.04.2025).
6. FasterCapital – The Rise of Price Comparison Apps. URL: <https://fastercapital.com/topics/the-rise-of-price-comparisonapps.html#:~:text=Price%20comparison%20apps%20are%20a%20valuable%20tool%20for%20consumers%20looking,possible%20deal%20on%20a%20product> (дата звернення 11.04.2025).
7. IdeaUsher – Price Comparison App Development. URL: <https://ideausher.com/blog/price-comparison-app-development/> (дата звернення 13.04.2025).
8. Shaikh, A., Khan, R., Panokher, K., Ranjan, M., & Sonaje, V. (2023). 'Ecommerce Price Comparison Website Using Web Scraping. *International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences*, 11, 1-13.
9. Singh, N., Rana, A., & Chaudhary, A. (2023, October). Price Comparison Using Web Scraping and Machine Learning. In *2023 International Conference on Computer Science and Emerging Technologies (CSET)* (pp. 1-5). IEEE.
10. ProIT – Як вебскрейпінг допомагає збирати дані. URL: <https://proit.org.ua/iak-viebskrieipingh-dopomaghaie-zbirati-dani-osnovni-tiekhnologhiyi-bibliotieki-ta-instrumenti-dlia-viebskrieipinghu/> (дата звернення 14.04.2025).

11. Bartók, O., Kozák, V., & Bauerová, R. (2021). Online grocery shopping: The customers perspective in the Czech Republic. *Equilibrium. Quarterly Journal of Economics and Economic Policy*, 16(3), 679-695.
12. Tan, C. H., Goh, K. Y., & Teo, H. H. (2010). Effects of comparison shopping websites on market performance: Does market structure matter? *Journal of Electronic Commerce Research*, 11(3).
13. Medium – On-Demand Grocery Delivery App Case Study. URL: <https://medium.com/the-gradient/design-case-study-on-demand-grocery-delivery-app-276fce99f386> (дата звернення 17.04.2025).
14. Bellini, S., & Aiolfi, S. (2017). The impact of mobile device use on shopper behaviour in store: An empirical research on grocery retailing. *International Business Research*, 10(4), 58-68.
15. DBeaver. URL: <https://dbeaver.io/> (дата звернення 22.04.2025).
16. SQLite. URL: <https://sqlite.org/> (дата звернення 22.04.2025).
17. SQLiteStudio. URL: <https://sqlitestudio.pl/> (дата звернення 22.04.2025).
18. Творошенко, І.С. (2023). Об'єктно-орієнтоване моделювання комп'ютерних систем: навч. посібник. Харків: ХНУРЕ.
19. Tvoroshenko I., and Gorokhovatskyi V. (2022) The Application of Hybrid Intelligence Systems for Dynamic Data Analysis, *International Journal of Engineering and Information Systems*, 6(2), pp. 40-48.
20. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Development of an application for recognizing emotions using convolutional neural networks, *International Journal of Academic Information Systems Research*, 7(7), pp. 25-36.
21. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2024) Improving the effectiveness of image classification structural methods by compressing the description according to the information content criterion, *Computers, Materials & Continua*, vol. 80, no. 2, pp. 3085-3106.

22. Gorokhovatskyi V., Tvoroshenko I., and Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 1, pp. 113-125.

23. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Handwritten character recognition models based on convolutional neural networks, *International Journal of Academic Engineering Research*, 7(9), pp. 64-72.

24. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, *Сучасні інформаційні системи*, 7(1), С. 5-13.

25. Gorokhovatskyi, V., Tvoroshenko, I., Kobylin, O., & Vlasenko, N. (2023). Search for visual objects by request in the form of a cluster representation for the structural image description, *Advances in Electrical and Electronic Engineering*, 21(1), pp. 19-27.

26. Gorokhovatskyi V., Chmutov Y., Tvoroshenko I., and Kobylin O. (2025) Reducing computational costs by compressing the structural description in image classification methods, *Advanced Information Systems*, vol. 9, no. 1, pp. 5–12.

27. Tvoroshenko I., Gorokhovatskyi V., Kobylin O., and Tvoroshenko A. (2023) Application of deep learning methods for recognizing and classifying culinary dishes in images, *International Journal of Academic and Applied Research*, 7(9), pp. 57-70.

28. Tvoroshenko I., Pomazan V., Gorokhovatskyi V., and Kobylin O. (2023) Application of video data classification models using convolutional neural networks, *International Journal of Academic and Applied Research*, 7(11), pp. 134-145.

29. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Gadetska S., and Al-Dhaifallah M. (2023) Statistical data analysis models for determining the relevance of structural image descriptions, *IEEE Access*, vol. 11, pp. 126938-126949.

30. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. *International scientific symposium «INTELLIGENT SOLUTIONS-S». Computational intelligence (results, problems and perspectives). Decision making theory: proceedings of the international symposium, September 28, 2023, Kyiv-Uzhorod, Ukraine*, pp. 25-27.

31. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., Hudáková M., and Gorokhovatskyi O. (2024) Application a committee of Kohonen neural networks to training of image classifier based on description of descriptors set, *IEEE Access*, vol. 12, pp. 73376-73385.

32. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., and Hudáková M. (2025) Image description compression in classification structural methods, *IEEE Access*, vol. 13, pp. 43631-43641.

33. Yakovleva O., Matúšová S., Tvoroshenko I., and Isaiev Y. (2024) Visitor counting based on video stream analysis from surveillance cameras to solve various business problems, *Verejná správa a regionálny rozvoj ekonómia, manažment a marketing*, XX(1), pp. 67-87.

34. Увохова І.С. (2025) Особливості сучасних вебзастосунків для порівняння цін, пошуку та формування списку замовлених товарів. *Радіoeлектроніка і молодь у XXI столітті: тези доповідей 29-го Міжнародного молодіжного форуму (Харків, 16–19 квітня 2025 р.)*. Харків: ХНУРЕ, 2025. Т. 7. С. 175-177.