

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра інженерії програмного забезпечення

### КВАЛІФІКАЦІЙНА РОБОТА

Садовського Артема Олександровича

Прізвище, ім'я, по батькові студента

На здобуття ступеня вищої освіти Бакалавра

«Вебзастосунок для реалізації молочної продукції»

Назва теми

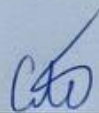
Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

Шифр КвРІПЗ. 2101088.01.15.ПЗ

Виконав студент IV курсу групи ПЗ-21-1



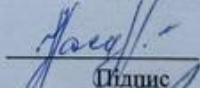
Підпис

Артем САДОВСЬКИЙ

Ім'я, прізвище

Керівник асистент

Науковий ступінь, звання

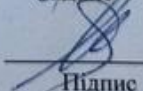


Підпис

Анастасія ДЬОМІНА

Ім'я, прізвище

Нормоконтролер канд. техн. наук, доцент



Підпис

Оксана ЯШИНА

Ім'я, прізвище

**До захисту допускаю:**

Завідувач кафедри інженерії  
програмного забезпечення



Підпис

Леонід БЕДРАТЮК

Ім'я, прізвище

10 червня 2025 р.

Хмельницький 2025

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Інженерії програмного забезпечення

Рівень вищої освіти Перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри ІПЗ

Л. П. Бедратюк

02 01 2025 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Садовському Артему Олександровичу

Прізвище, ім'я, по батькові студента

1. Тема роботи Вебзастосунок для реалізації молочної продукції

Керівник роботи Дьоміна Анастасія Іванівна, асистент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 08.01.2025 р. №6-КП

2. Строк подання студентом роботи на кафедру 01.06.25 р.

3. Вихідні дані до роботи Методичні матеріали до кваліфікаційної роботи

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Дослідження предметної області та постановка задачі, проєктування програмного забезпечення, програмна реалізація, тестування онлайн-ресурсу

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Три креслення:

1. Діаграма варіантів використання

2. Схеми бази даних

3. Архітектурні рішення

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Яшина О. М., доцент кафедри ІПЗ	08.04.2025	06.06.2025
Антиплагіат	Форкун Ю. В., доцент кафедри ІПЗ	12.01.2025	09.06.2025

7. Дата видачі завдання « 02 » січня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1 Ознайомлення з тематикою кваліфікаційної роботи (КвР), визначення та узгодження індивідуальних тем КвР	01.12 – 31.12.2024	
2 Збір матеріалу за темою КвР; дослідження предметної області, в якій планується використання програмного забезпечення (ПЗ), визначення задач та вимог, розробка технічного завдання	02.01 – 31.01.2025	
3 Проектування програмного забезпечення	01.02 – 28.02.2025	
4 Програмна реалізація з використанням відповідних засобів розробки	01.03 – 10.04.2025	
5 Тестування програмного забезпечення	11.04 – 30.04.2025	
6 Написання вступу, загальних висновків, оформлення переліку джерел посилання та додатків. Оформлення пояснювальної записки КвР згідно вимог	01.05 – 25.05.2025	
7 Попередній захист КвР	травень 2025 (згідно графіка)	
8 Перевірка КвР на плагіат, нормоконтроль, отримання відгуків, рецензій та інших супровідних документів. Брошурування (зшиття) пояснювальної записки.	26.05 – 30.05.2025	
9 Здача КвР на кафедру; підготовка КвР для розміщення у репозиторії ХНУ; підготовка до захисту та захист КвР	з 01.06.2025	

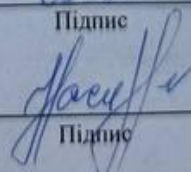
Студент

  
Підпис

Артем САДОВСЬКИЙ

Ініціали, прізвище

Керівник роботи

  
Підпис

Анастасія ДЬОМІНА

Ініціали, прізвище

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Вебзастосунок для реалізації молочної продукції».

Автор роботи: Садовський Артем Олександрович.

Керівник роботи: Дьоміна Анастасія Іванівна.

Пояснювальна записка: 84 с., 26 рис., 5 табл., 3 дод., 40 джерел.

Графічна частина: 3 креслення у форматі А3.

ВЕБЗАСТОСУНОК, ОНЛАЙН-РЕСУРС, ФРЕЙМВОРК, МОВА ПРОГРАМУВАННЯ, ОНЛАЙН-ПЛАТФОРМА, ЕЛЕКТРОННА КОМЕРЦІЯ, МОЛОКОПРОДУКТИ, ПОКУПКА ОНЛАЙН, МЕРЕЖА ІНТЕРНЕТ, MVC, SQL, HTML, CSS, C#.

Метою кваліфікаційної роботи є розробка вебзастосунку для продажу молочної продукції, що має можливість реалізувати функціонал різній категорії користувачів.

Під час проведення роботи було здійснено аналіз предметної області, а саме інтернет-торгівлі молочними продуктами, використовуючи відповідні джерела у всесвітній павутині. Також було визначено вимоги до розроблюваного вебзастосунку, вказано завдання, після вирішення яких буде досягнуто поставленої мети; описано проектування бази даних; подано опис та вибір архітектури розроблюваного програмного продукту.

Результатом даної кваліфікаційної роботи є вебзастосунок для реалізації молочної продукції, при реєстрації на якому користувач отримує доступ до повноцінного каталогу, пошуку товару, замовлення, знижок, здійснення покупки, замовлення із оплатою та типом доставки.

02.06.2025

Дата

САД

Підпис

## ВІДОМІСТЬ ДОКУМЕНТІВ

№ рядка	Формат	Позначення документа	Найменування документа	К-сть аркушів	№ екз.	Примітка
			Текстові документи			
1	A4	КвРІПЗ.2101088.01.15.П	Пояснювальна записка	84		
2	A4		Завдання на кваліфікаційну роботу	2		
3	A4		Анотація	1		
			Графічні документи			
4	A3	КвРІПЗ.2101088.01.15.E8	Діаграма варіантів використання	1		
5	A3	КвРІПЗ.2101088.01.15.E8	Схема Баз даних	1		
6	A3	КвРІПЗ.2101088.01.15.E8	Архітектура	1		

КвРІПЗ. 2101088.01.15.ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата
Виконав		Садовський	<i>[Signature]</i>	02.07
Керівник		Дьоміна А.І.	<i>[Signature]</i>	02.06
Н. контр.		Яшина О.М.	<i>[Signature]</i>	02.06
Зав. каф.		Бедратюк Л.П.	<i>[Signature]</i>	02.06
			Вебзастосунок для реалізації молочної продукції	Літ.
			Відомість документів	Арк.
				Аркушів
				5
				84
ХНУ, ІПЗ-21-1				

## ЗМІСТ

ВСТУП.....	7
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ....	9
1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей.....	9
1.2 Аналіз наявного програмно-технічного забезпечення предметної області .....	11
1.3 Визначення функціональних та нефункціональних вимог до програмного забезпечення. ....	18
1.4 Висновки. Постановка задачі.....	22
2 ПРОЕКТУВАННЯ ВЕБЗАСТОСУНКУ .....	24
2.1 Проектування архітектури та структури .....	24
2.2 Детальне проектування програмного забезпечення .....	27
2.3 Проектування інтерфейсу користувача .....	33
2.4 Аналіз та вибір технологій і методів реалізації вебзастосунок.....	37
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ОНЛАЙН-РЕСУРСУ .....	42
3.1 Реалізація бази даних.....	42
3.2 Реалізація модулів вебзастосунок.....	45
3.3 Реалізація інтерфейсу. Інструкція користувача .....	46
3.4 Тестування вебзастосунок.....	52
ВИСНОВКИ .....	57
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	58
ДОДАТОК А Презентаційні матеріали .....	62
ДОДАТОК Б Технічне завдання .....	77

КвРІПЗ. 2101088.01.15.ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата
Виконала		Садовський	<i>С. Садовський</i>	20/06/20
Керівник		Дьоміна А.І.	<i>А. Дьоміна</i>	20/06/20
Н. контр.		Яшина О.М.	<i>О. Яшина</i>	20/06/20
Зав. каф.		Бедратюк Л.П.	<i>Л. Бедратюк</i>	20/06/20
Вебзастосунок для реалізації молочної продукції				
Відомість документів				
		Літ.	Арк.	Аркушів
			6	84
ХНУ, ІПЗ-21-1				

## ВСТУП

Молочна продукція – це специфічний вид продуктів харчування, що є одними з популярних серед населення не тільки України, але й світу. Відомо, що молоко виконує функцію захисного чинника для організму завдяки вмісту в ньому поживних речовин, головним чином, білка, лактози, мінеральних солей, вітамінів тощо [1].

Хоча електронна комерція може бути доступною не всім переробникам молочної продукції, вона безумовно може бути варіантом для отримання додаткового доходу. Минули ті часи, коли споживачі вагалися купувати продукти харчування та напої в Інтернеті, чому, безумовно, сприяла пандемія COVID-19.

Euromonitor International повідомила, що кількість американців, які користуються онлайн-сервісами продажу продуктів, різко зросла з 13% до пандемії до приблизно 26% у 2022 році. Згідно з кількома звітами, збільшення використання смартфонів призводить до зростання продажів електронної комерції [7].

Дані Marketing Charts аналогічні: близько 28% респондентів у 2022 році повідомили, що замовляють продукти онлайн для самовивезення або доставки принаймні раз на місяць, порівняно з 23% у попередньому році. Дослідницька фірма додає, що онлайн-покупки продуктів частіше зустрічаються серед домогосподарств з вищим, ніж з низьким рівнем доходу: 37% респондентів з річним доходом домогосподарства (ННІ) щонайменше 100 000 доларів США повідомили, що роблять це принаймні щомісяця, що приблизно вдвічі більше, ніж у тих, чий річний ННІ менше 40 000 доларів США (19%) [7].

Продажі в електронній комерції продуктів харчування та напоїв (F&V) у вигляді споживчих товарів (CPG) досягли 75,1 млрд доларів у 2023 році, а за останні 52 тижні, що закінчилися 11 серпня, вони досягли 80,1 млрд доларів, стверджує чиказька дослідницька компанія Circana. Це становить 9,2% та 9,6%

					КВРІПЗ.2101088.01.15.ПЗ	Арк.
						7
Зм.	Арк	№ докум.	Підпис	Дата		

від загального обсягу омніканальних продажів відповідно, порівняно з 7,4% у 2021 році [7].

Дані Circana також показують, що продажі в електронній комерції продовжують зростати з року в рік, навіть попри те, що пандемія вже позаду. «Споживачі продовжують орієнтуватися на цінність та зручність, і ціннісна пропозиція електронної комерції це забезпечує», – стверджує Енн Зибовскі, віцепрезидент з електронної комерції чиказької компанії Circana. «Понад 84% покупців – це омніканальні покупці, і навіть якщо покупці не купують онлайн, досвід електронної комерції впливає на понад 70% продажів».

Метою кваліфікаційної роботи є розробка вебзастосунку для продажу молочної продукції, що має можливість реалізувати функціонал різній категорії користувачів.

Процес розробки передбачає вирішення таких завдань:

- проаналізувати ринок реалізації молочної продукції в мережі Інтернет;
- провести аналітичний огляд існуючих програмно-технічних рішень, а саме вебресурсів з продажу молокопродуктів із виділенням негативних та позитивних практик реалізації;
- визначити функціональні та нефункціональні вимоги розроблюваного програмного застосунку;
- розробити технічне завдання;
- здійснити вибір типу архітектури та проектування бази даних;
- здійснити реалізації програмного застосунку для реалізації молочної продукції; протестувати вебзастосунок для реалізації молочної продукції.

					КВРІПЗ.2101088.01.15.ПЗ	Арк.
						8
Зм.	Арк	№ докум.	Підпис	Дата		

# 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей

Аналіз предметної області є відправною точкою для подальшого детального проектування вебзастосунку. Він допоможе команді розробників, дизайнерам та замовнику чітко зрозуміти обсяг робіт, основні вимоги та очікувані результати.

Вебзастосунок – це інтерактивна програма, яка працює на сервері та доступ до якої здійснюється через веб-браузер. Такі застосунки необхідні для надання послуг користувачам і пропонують цінний зворотний зв'язок розробникам. Такий зворотний зв'язок часто включає дані про використання, що надають уявлення про взаємодію користувачів, уподобання та часто використовувані функції.

Вебзастосунки важливі, оскільки вони покращують взаємодію з користувачами та надають компаніям критично важливі дані для формування стратегій розробки продуктів. Компанії можуть налаштовувати свої пропозиції, аналізуючи поведінку користувачів, покращуючи продуктивність та узгоджуючи свої послуги з очікуваннями клієнтів.

Вебзастосунки залежать від кількох основних технічних компонентів для безперебійного функціонування. Розуміння цих елементів має вирішальне значення для створення переконливих вебзастосунків та забезпечення безперебійної взаємодії користувачів із програмним забезпеченням. До структурних елементів роботи вебзастосунку відноситься розробка на стороні сервера, розробка на стороні клієнта, керування базами даних

Розробка на стороні клієнта зосереджена на створенні того, що бачить користувач і з чим взаємодіє. Такі технології, як HTML, CSS та JavaScript, дозволяють розробникам створювати адаптивні, візуально привабливі та динамічні інтерфейси. Клієнтську сторону часто називають «інтерфейсом»,

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		9

оскільки вона відповідає за зовнішній вигляд та поведінку програми, представленої кінцевому користувачеві. Деякі популярні фреймворки для розробки на стороні клієнта включають React, Angular, Vue.js та Svelte.

Під час розробки на стороні сервера потрібно враховувати той факт, що сервер виконує автентифікацію користувачів, управління базами даних та бізнес-логіку. Поширені серверні мови включають Python, Ruby, PHP та Java. Саме ці технології бекенду забезпечують швидку та безпечну обробку запитів front-end.

Бази даних зберігають усі дані веб-програми. Розробники зазвичай використовують системи на основі SQL або NoSQL, такі як MongoDB, для забезпечення швидкої обробки та обробки даних. Вебпрограми зазвичай потребують добре структурованих баз даних для ефективного управління даними та інших структур.

Розробники відповідають за проектування та впровадження схем баз даних, створення запитів для отримання та маніпулювання даними, а також за забезпечення цілісності та безпеки даних. Такий структурований підхід допомагає підтримувати стабільний та надійний доступ до даних, що є вирішальним для загальної продуктивності програми.

Маючи базове розуміння про вебзастосунки та основних компонентів, які ними керують, можна стверджувати, що вебзастосунки бувають різних форм, кожна з яких розроблена для задоволення конкретних потреб користувачів та покращення взаємодії з ними. Розуміння цих типів може допомогти компаніям вибрати правильне рішення для своїх цілей.

Отже, основною метою розроблюваного вебзастосунку є створення ефективного онлайн-каналу для продажу молочної продукції, що дозволить виробникам або постачальникам розширити клієнтську базу, оптимізувати процеси замовлення та доставки, а споживачам - зручно купувати свіжі та якісні молочні продукти.

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
						10
Зм.	Арк	№ докум.	Підпис	Дата		

## 1.2 Аналіз наявного програмно-технічного забезпечення предметної області

Оскільки у дипломній кваліфікаційній роботі передбачено проектування та розробку вебзастосунку для продажу молочної продукції, то було проаналізовано існуючі ресурси та виділено їх переваги та недоліки, щоб врахувати це у подальшій роботі.

### 1. Інтернет-магазин «Галичина» [5]

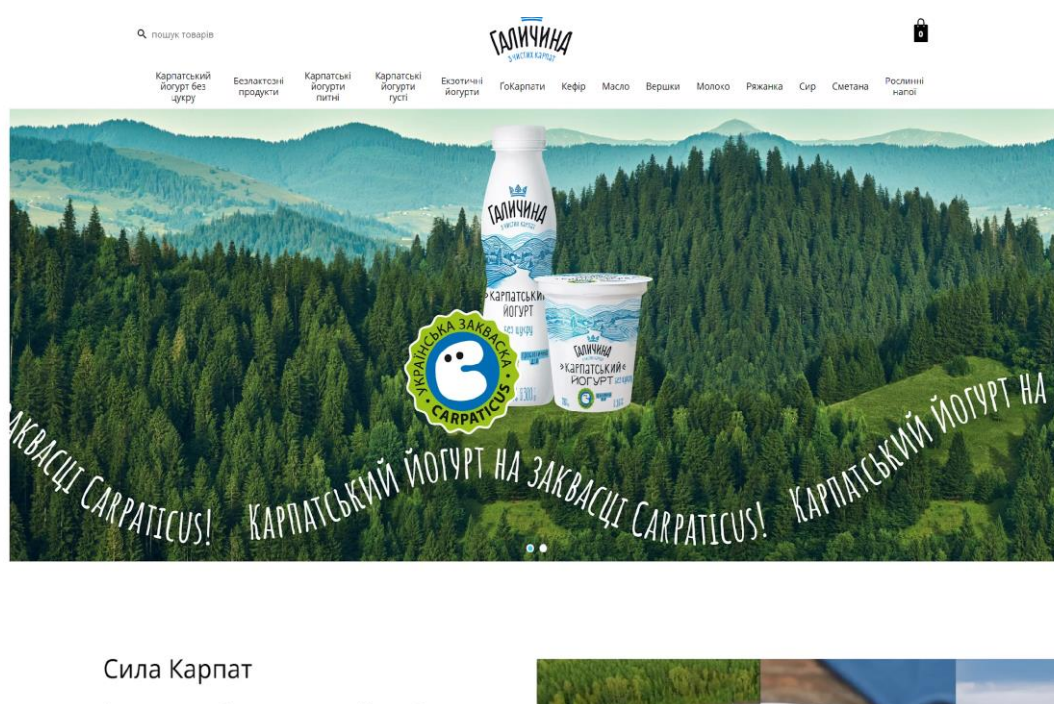


Рисунок 1.1 – Головна сторінка ресурсу [5]

#### Переваги:

- чистий та сучасний дизайн;
- акцент на зображеннях продукції;
- зручна навігація по категоріях.

#### Недоліки:

- може бути трохи перевантажений інформацією на головній сторінці;
- деяким користувачам може не вистачати більш детальної інформації про виробника на сторінках продуктів;

					КВРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		11

Розробка:

Переваги:

- швидке завантаження сторінок;
- адаптивний дизайн (добре виглядає на мобільних);
- інтеграція з платіжними системами.

Недоліки:

Складно сказати без детального аналізу коду, але можливо, використовують стандартну e-commerce платформу, що може обмежувати кастомізацію.

## 2. Інтернет-магазин «Sladosvit» [6]

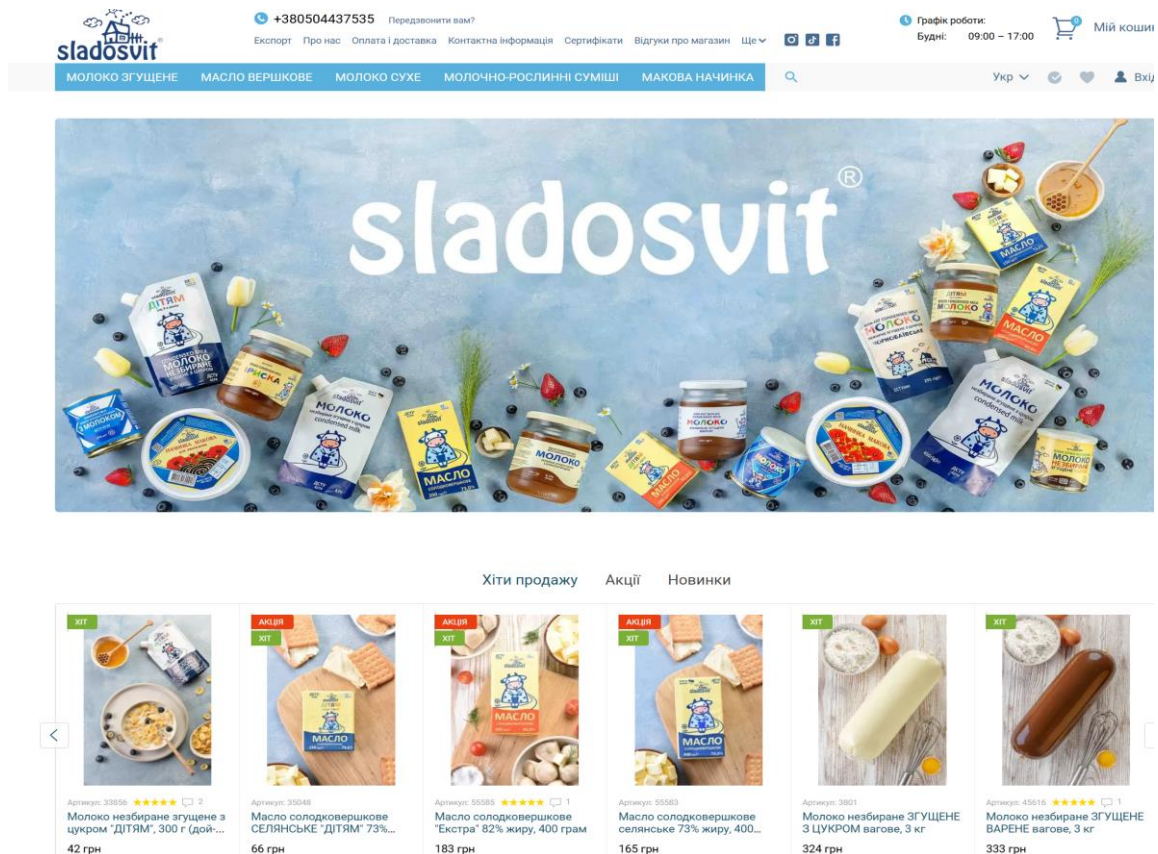


Рисунок 1.2 – головна сторінка [6]

Переваги:

- простий та зрозумілий дизайн;
- акцент на натуральності продукції;

					КВРІПЗ. 2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		12

- легкість в обслуговуванні та оновленні;
- можливо, використовують просту CMS;
- інформація про виробника на видному місці.

Недоліки:

- дизайн може здаватися трохи застарілим;
- обмежена функціональність фільтрації та сортування;
- обмежена масштабованість;
- можливі проблеми з безпекою, якщо не оновлюється регулярно.

### 3. «Інтернет ринок Шувар» [7]

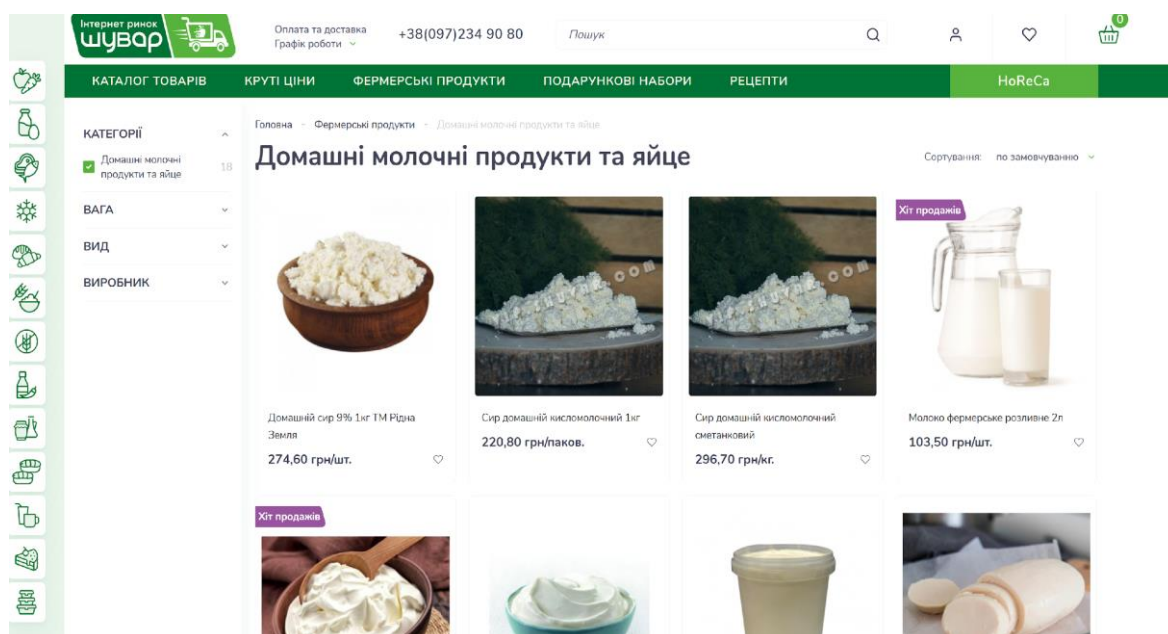


Рисунок 1.3 – Розділ молочної продукції «Інтернет ринок Шувар»

Переваги:

- акцент на фермерській продукції;
- інформація про виробників;
- можливість підтримки різних продавців.

Недоліки:

- загальний дизайн сайту може бути перевантаженим;

					КвРІПЗ. 2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		13

- навігація може бути не зовсім інтуїтивною;
- складність підтримки та оновлення великого маркетплейсу;
- різна якість коду від різних продавців.

#### 4. «Molokoferma» [8]

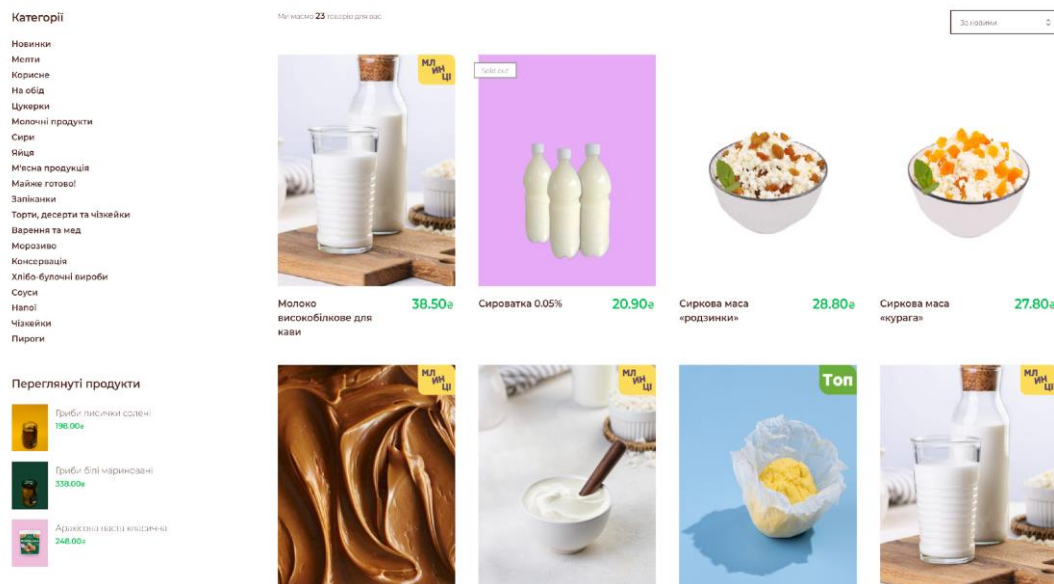


Рисунок 1.4 – Головна сторінка ресурсу «Molokoferma»

До гарних сторінок даної розробки можна віднести сучасний та мінімалістичний дизайн та зручна навігація.

#### Недоліки:

- може здаватися дещо «холодним» для деяких користувачів.

#### Переваги:

- швидка робота сайту;
- використання сучасних технологій;
- обмежені можливості кастомізації.

Успішні сайти з продажу молочної продукції часто мають чистий, сучасний дизайн з акцентом на якісних зображеннях продукції. Важлива зручна навігація та зрозуміла структура каталогу.

					КВРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		14

Використовуються різні підходи, від простих CMS до складних e-commerce платформ. Важлива швидкість завантаження, адаптивність, безпека та можливість інтеграції з платіжними системами та службами доставки. Масштабованість є ключовим фактором для великих магазинів.

Ці приклади ілюструють різноманітні підходи до дизайну та розробки вебсайтів з продажу молочної продукції. Вибір технологій та дизайну залежить від цільової аудиторії, розміру бізнесу та бюджету.

## 5. Фермерський магазин [9]

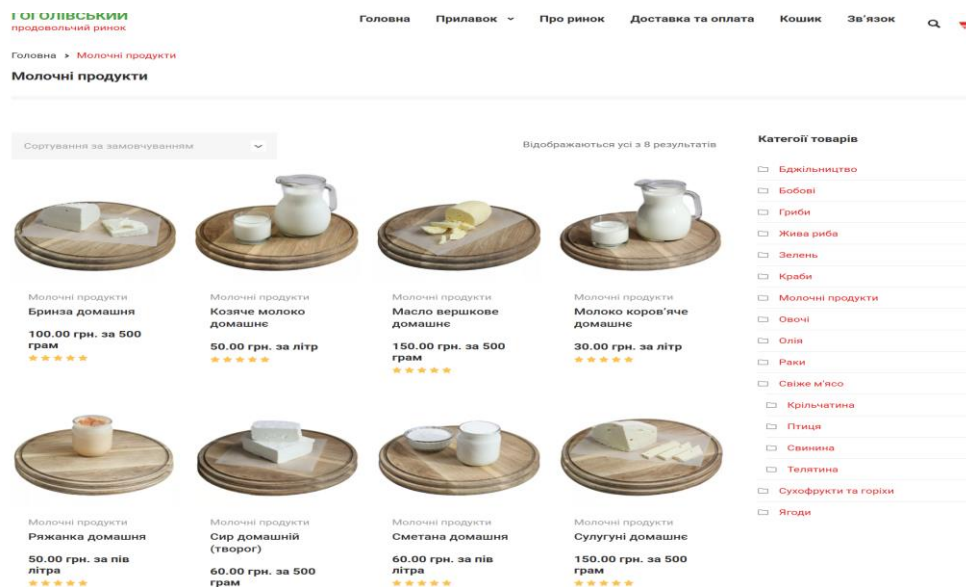


Рисунок 1.5 – Головна сторінка ресурсу [9]

### Переваги:

- дизайн у стилі «еко» або «фермерський», що підкреслює натуральність продукції;
- акцент на фотографіях, що демонструють якість та свіжість продуктів;
- наявність інформації про ферму та виробництво, що підвищує довіру.

### Недоліки:

- може здаватися дещо простим або не таким сучасним, як у великих

– навігація може бути менш інтуїтивною для великого асортименту (якщо він розширюватиметься).

Переваги:

– можливо, використовують просту cms (наприклад, wordpress з woocommerce), що полегшує управління контентом та товарами;

– порівняно невисока вартість розробки та підтримки на початковому етапі.

Недоліки:

– масштабованість може стати проблемою при значному зростанні асортименту або трафіку;

– залежність від плагінів може впливати на стабільність та безпеку.

6. Мережа магазинів «VOVA», що займається доставкою молокопродуктів для кафе, ресторанів, барів.

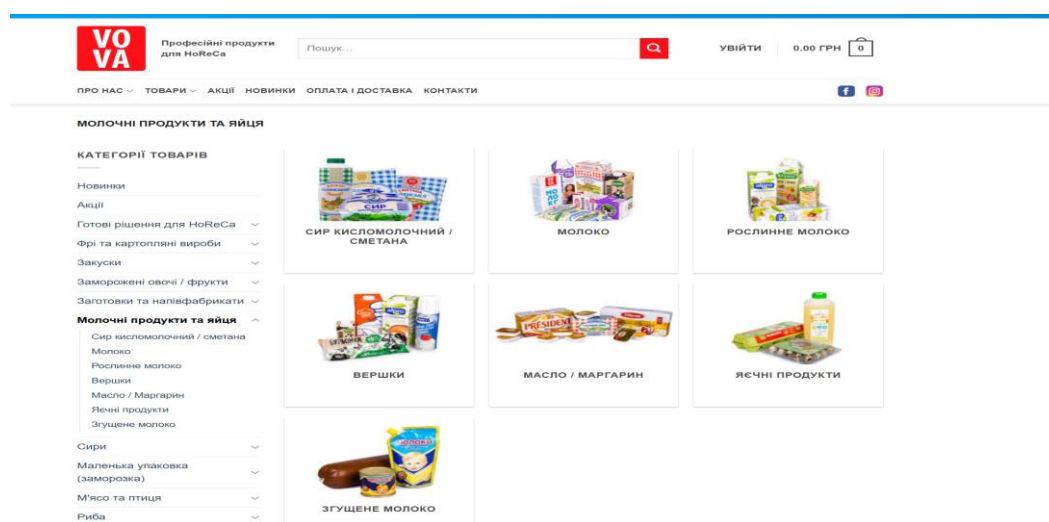


Рисунок 1.6 – Головна сторінка ресурсу «VOVA»

До переваг можна віднести простий, орієнтований на швидке замовлення та контактну інформацію. Може підкреслювати місцеве виробництво.

					КвРІПЗ. 2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		16

До недоліків відносимо неконсистентним дизайн між різними точками мережі (якщо вони мають окремі онлайн-представництва). Функціональність онлайн-замовлення може бути обмеженою.

Перевагами є низькі витрати на розробку (можуть використовуватись прості лендінги або сторінки в соцмережах).

Недоліками є обмежена функціональність для онлайн-торгівлі, складнощі з інтеграцією платіжних систем та служб доставки, відсутність централізованої системи управління замовленнями.

7. Спеціалізовані крафтові виробники молочної продукції (часто представлені в маркетплейсах типу або мають власні невеликі сайти)

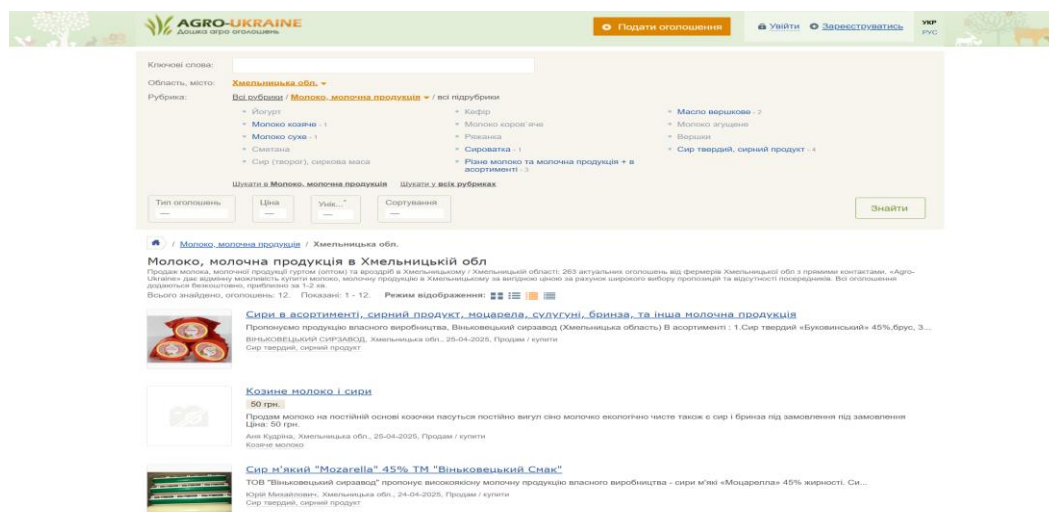


Рисунок 1.7 – Приклад інтерфейсу маркетплейсу для виробників молочної продукції

Переваги:

- унікальний, часто ручної роботи дизайн, що підкреслює ексклюзивність продукції. акцент на історії виробництва та майстрах;
- можливість використання креативних технологій та індивідуальних рішень.

Недоліки:

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		17

– може бути менш орієнтованим на масового споживача, навігація може бути не завжди стандартною;

– може бути дорожчим у розробці та підтримці, особливо при складних кастомних рішеннях. Масштабованість може бути обмеженою.

Отже, аналіз наявного програмного забезпечення показує, що в існуючих рішеннях уже реалізовано базовий функціонал, необхідний для роботи в даній предметній області. Разом з тим, у більшості з них спостерігаються певні обмеження або недоліки, що відкриває можливості для подальшого вдосконалення та розробки більш ефективного рішення.

### 1.3 Визначення функціональних та нефункціональних вимог до програмного забезпечення

Проаналізувавши предметну область та уже існуючі рішення, необхідно здійснити аналіз та обґрунтування вимог до розроблюваного програмного продукту, оскільки саме від цього залежить якість виконуваної розробки.

Під час аналізу вимог прийнято виділяти функціональні та нефункціональні вимоги. Функціональні вимоги - визначають конкретні дії, які повинна виконувати система. Нефункціональні вимоги - описують, як система повинна працювати. Вони визначають критерії, якість або обмеження, яким повинна відповідати система.

Нижче подано визначені функціональні та нефункціональні вимоги, що висуваються до розробки вебзастосунку для продажу молочної продукції.

Функціональні вимоги визначають, що система повинна містити каталог продукції, обробка замовлень, керування обліковим записом користувача, адміністрування (для адміністраторів магазину), додаткові функції.

Детальний перелік кожного з цих пунктів подано нижче.

Каталог продукції має містити:

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		18

- перегляд списку молочних продуктів (молоко, сир, йогурт тощо);
- розбиття на категорії (наприклад, за типом продукту, брендом);
- розширений пошук і фільтрація (за назвою, категорією, харчовою цінністю, алергенами);
- детальні сторінки продукту з описом, зображеннями, інформацією про харчову цінність та відгуками.

Обробка замовлень передбачає:

- кошик для додавання, редагування та видалення товарів;
- процес оформлення замовлення з вибором адреси доставки та способу оплати;
- підтвердження замовлення з деталями та номером відстеження;
- історія замовлень для користувачів.

Керування обліковим записом користувача має містити:

- реєстрацію та вхід в обліковий запис;
- редагування профілю (адреса, контактні дані);
- збереження улюблених товарів;
- підписку на розсилку новин.

Адміністрування (для адміністраторів магазину) має містити:

- керування каталогом продукції (додавання, редагування, видалення продуктів);
- керування замовленнями (перегляд, оновлення статусу, обробка відшкодувань);
- керування користувачами;
- аналітику та звіти (продажі, популярні товари).

Додаткові функції можуть передбачати:

- відгуки та рейтинги товарів;
- акції та знижки;
- інтеграцію з платіжними системами;
- службу підтримки клієнтів (чат, форма зворотного зв'язку).

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
						19
Зм.	Арк	№ докум.	Підпис	Дата		

Нефункціональні вимоги відповідають на питання як повинна працювати система і передбачають такі характеристики як продуктивність, безпека, зручність використання, надійність, масштабованість, підтримка. Складові даних властивостей розписано нижче.

**Продуктивність:**

- швидке завантаження сторінок;
- плавна навігація;
- обробка великої кількості одночасних користувачів.

**Безпека:**

- захист даних користувачів (GDPR);
- безпечні платежі (PCI DSS);
- захист від поширених веб-атак (XSS, SQL-ін'єкції).

**Зручність використання:**

- інтуїтивно зрозумілий інтерфейс;
- зручна навігація;
- адаптивний дизайн (для мобільних пристроїв).

**Надійність:**

- мінімальний час простою;
- регулярне резервне копіювання даних;
- обробка помилок.

**Масштабованість:**

- можливість обробки зростаючої кількості товарів і користувачів;
- легке додавання нових функцій.

**Підтримка:**

- детальна документація для користувачів та адміністраторів;
- можливість обслуговування та оновлення.

На основі описаних вимог було сформовано діаграму варіантів використання для зареєстрованого користувача та адміністратора (рис.1.8-1.9).

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		20

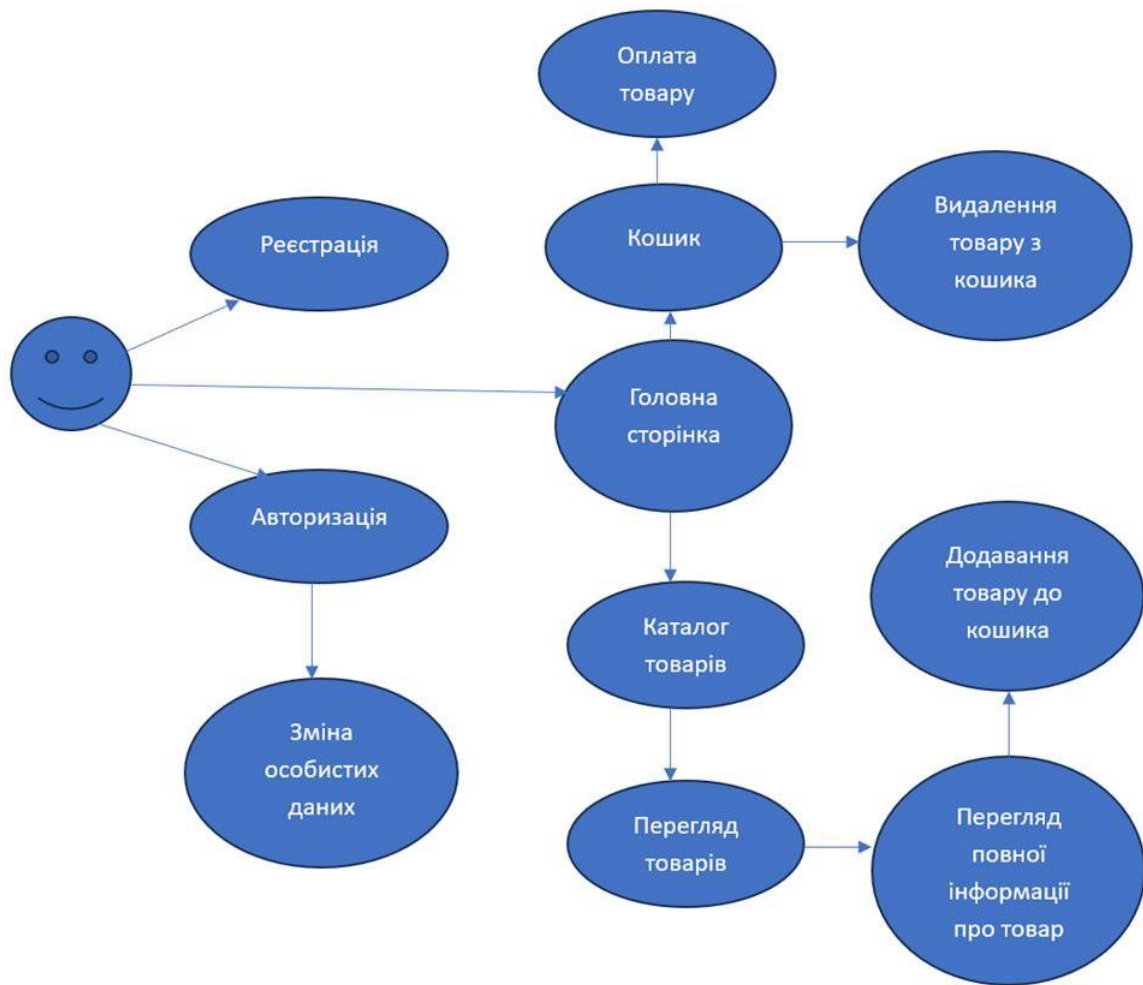


Рисунок 1.8 – Діаграма варіантів використання для зареєстрованого користувача

#### 1.4 Висновки. Постановка задачі

Отже, для розробки вебзастосунку з реалізації молочної продукції було здійснено аналіз предметної області та встановлено, що вебзастосунком називають інтерактивну програму, яка працює на сервері та доступ до якої здійснюється через веббраузер. Такі застосунки необхідні для надання послуг користувачам і пропонують цінний зворотний зв'язок розробникам. Такий зворотний зв'язок часто включає дані про використання, що надають уявлення про взаємодію користувачів, уподобання та часто використовувані функції.

В даному розділі було проаналізовано існуючі програмно-технічні рішення та виявлено, що для якісної та ефективної роботи вебзастосунку необхідно врахувати переваги та недоліки даних ресурсів. Зокрема дотримуватись мінімалістичного дизайну інтерфейсу, врахувати необхідність пошуку інформації, встановлення каталогів продукції, можливість перегляду інформації за певним продуктом, забезпечити масштабованість ресурсу тощо.

Загалом передбачається, що ресурс має містити каталог продукції, що повинен містити:

- перегляд списку молочних продуктів;
- розбиття на категорії;
- розширений пошук і фільтрацію;
- детальні сторінки продукту з описом.

Обробка замовлень передбачає:

- кошик для додавання, редагування та видалення товарів;
- процес оформлення замовлення з вибором адреси доставки та способу оплати;
- підтвердження замовлення з деталями та номером відстеження;
- історія замовлень для користувачів.

Ресурс повинен передбачати вхід за ролями адміністратор, зареєстрований, незареєстрований користувач.

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		22

Керування обліковим записом користувача передбачає:

- реєстрацію та вхід в обліковий запис;
- редагування профілю (адреса, контактні дані);
- збереження улюблених товарів;
- підписку на розсилку новин.

Адміністрування передбачає:

- керування каталогом продукції (додавання, редагування, видалення продуктів);
- керування замовленнями (перегляд, оновлення статусу, обробка відшкодувань);
- керування користувачами;
- аналітику та звіти (продажі, популярні товари).

Додаткові функції можуть містити:

- відгуки та рейтинги товарів;
- акції та знижки;
- інтеграцію з платіжними системами;
- службу підтримки клієнтів (чат, форма зворотного зв'язку).

					КвРІПЗ. 2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		23

## 2 ПРОЕКТУВАННЯ ВЕБЗАСТОСУНКУ

### 2.1 Проектування архітектури та структури системи

Для проекту було обрано клієнт-серверну архітектуру – розподілену архітектуру застосунків, яка розподіляє завдання або робоче навантаження між серверами (постачальниками ресурсів або послуг) і клієнтами (запитувачами цих послуг). У цій моделі клієнт надсилає серверу запит на отримання даних, які зазвичай обробляються на стороні сервера. Потім сервер повертає запитані дані клієнту.

Клієнти, як правило, не діляться ресурсами один з одним, а покладаються на сервер, який надає запитовані ресурси або послуги. Поширеними прикладами клієнт-серверної моделі є системи електронної пошти та Всесвітня павутина (WWW), де поштові клієнти взаємодіють з поштовими серверами, а веббраузери запитують ресурси з веб-серверів.

Коли мова йде про «клієнта», то маємо на увазі пристрій (зазвичай комп'ютер, смартфон або додаток), який запитує і отримує послуги від сервера. Клієнт - це об'єкт, який ініціює зв'язок, запитуючи дані або ресурси від сервера. Наприклад, веб-браузери, такі як Google Chrome, Mozilla Firefox або Safari, є поширеними клієнтськими програмами, які запитують дані з сервера для відображення веб-сторінок.

З іншого боку, сервер - це віддалений комп'ютер або система, яка надає дані, ресурси або послуги клієнтам. Він прослуховує вхідні клієнтські запити, обробляє їх і надсилає необхідну інформацію назад. Сервер може обробляти кілька клієнтських запитів одночасно.

Наприклад, вебсервери розміщують вебсайти, а сервери баз даних зберігають і обслуговують бази даних для додатків. Простіше кажучи, клієнт надсилає запит на сервер, а сервер обслуговує його доти, доки дані або послуга доступні в його системі.

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		24

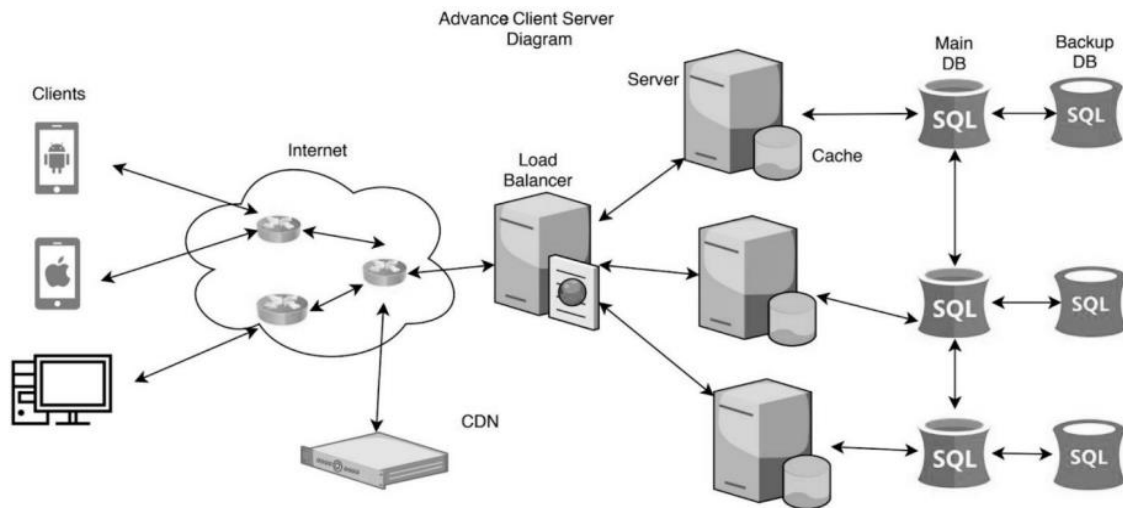


Рисунок 2.1 – Приклад клієнт-серверної архітектури

Процес взаємодії з серверами через браузер складається з декількох кроків. Нижче наведено розбивку кроків, які виконуються, коли вводиться URL-адреса в браузері та отримуються дані з вебсайту:

- користувач вводить URL-адресу (Uniform Resource Locator);
- пошук DNS (система доменних імен);
- DNS-сервер вирішує адресу;
- браузер надсилає HTTP/HTTPS запит;
- надсилання сервером файлів вебсайту;
- відображення вебсайту.

Користувач вводить адресу вебсайту (наприклад, [www.example.com](http://www.example.com)) в адресний рядок браузера. Браузер надсилає запит на DNS-сервер для перетворення людського URL-адреси в IP-адресу (оскільки комп'ютери використовують IP-адреси для ідентифікації та з'єднання один з одним). Далі DNS-сервер шукає доменне ім'я і повертає IP-адресу веб-сервера, на якому розміщений запитуваний вебсайт. Браузер надсилає HTTP/HTTPS-запит на IP-адресу вебсервера, щоб отримати дані вебсайту. HTTP (HyperText Transfer Protocol) або HTTPS (захищена версія) - це протокол, який використовується для зв'язку між браузером (клієнтом) і вебсервером (сервером).

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		25

Сервер обробляє запит і надсилає необхідні файли веб-сайту (HTML, CSS, JavaScript, зображення тощо) назад до браузера. Браузер відтворює файли і відображає вебсайт користувачеві. Цей процес рендерингу включає кілька компонентів: разом ці компоненти, відомі як компілятори Just-In-Time (JIT), дозволяють браузеру перетворити необроблені дані у візуальну веб-сторінку.

Проект реалізований як вебзастосунок на основі архітектури MVC (Model-View-Controller) за допомогою фреймворку Laravel. Схематично даний патерн продемонстровано на рисунку 2.2.

Основні компоненти, що входять до патерну MVC:

Моделі, тобто представлення даних та бізнес-логіки:

- Category - категорії продуктів;
- Product - молочні продукти;
- Customer - клієнти/користувачі системи;
- інші моделі для замовлень, оплат тощо.

Контролери відповідають за обробку запитів користувачів

- ProductController - управління продуктами;
- CategoryController - управління категоріями;
- CustomerAuthController - авторизація користувачів;
- Інші контролери для обробки специфічних запитів.

Види або представлення (Views) відповідають за відображення (рендеринг) даних:

- Blade шаблони для відображення контенту;
- Компоненти інтерфейсу користувача.

					КвРІПЗ. 2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		26

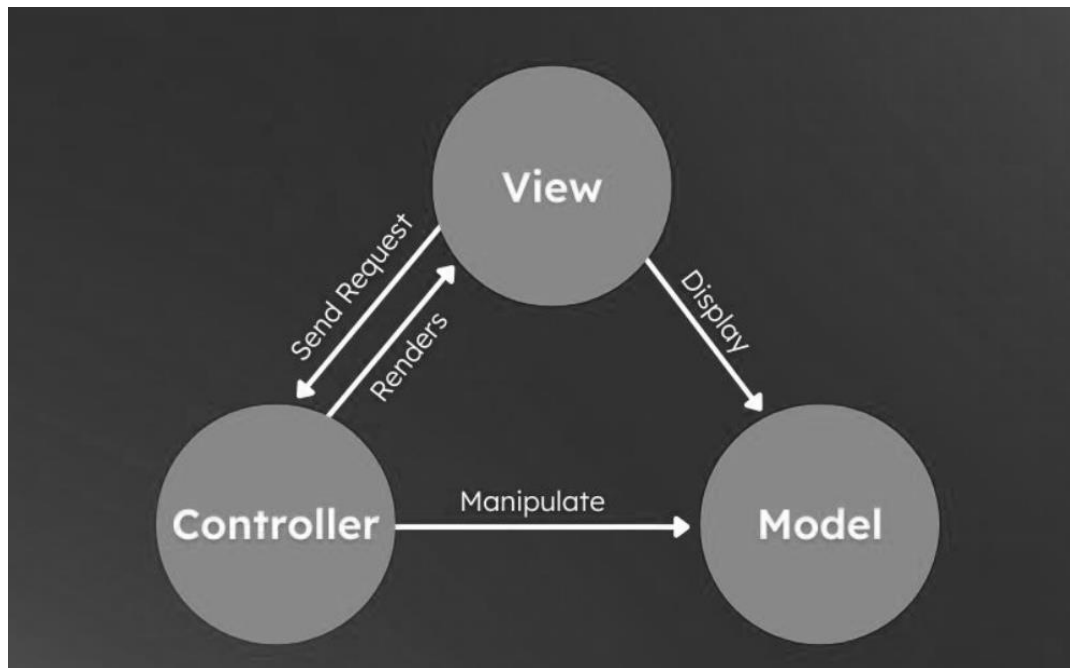


Рисунок 2.2 – Патерн MVC

## 2.2 Детальне проектування програмного забезпечення

Щоб визначити внутрішні властивості та деталізувати видимі компоненти застосовується такий комплекс заходів як проектування програмного забезпечення. Найкращим варіантом для деталізації даного етапу є моделювання.

Для ефективного аналізу функціональної структури вебзастосунку було використано модель IDEF0. Цей підхід дозволяє візуалізувати ключові процеси та взаємозв'язки між основними компонентами системи, забезпечуючи глибоке розуміння її функціонування. Рисунок 2.3 демонструє модель IDEF0.

Декомпозиція діаграми IDEF0 дозволила нам детально вивчити структуру та логіку роботи вебзастосунку для продажу молочної продукції, виявивши основні процеси та їх послідовність. Це допомагає оптимізувати роботу системи та покращити користувацький досвід. Рисунок 2.4 демонструє декомпозицію діаграми IDEF0.

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		27

Найбільш привабливим засобом для відображення та деталізації моделей є UML, а саме за допомогою діаграми компонентів, яка демонструє компоненти, що входять до системи, а також взаємозв'язки між ними.



Рисунок 2.3 – Діаграма IDEF0

Діаграми компонентів використовуються для представлення того, як організовані фізичні компоненти в системі. Їх доцільно використовувати для моделювання деталей реалізації.

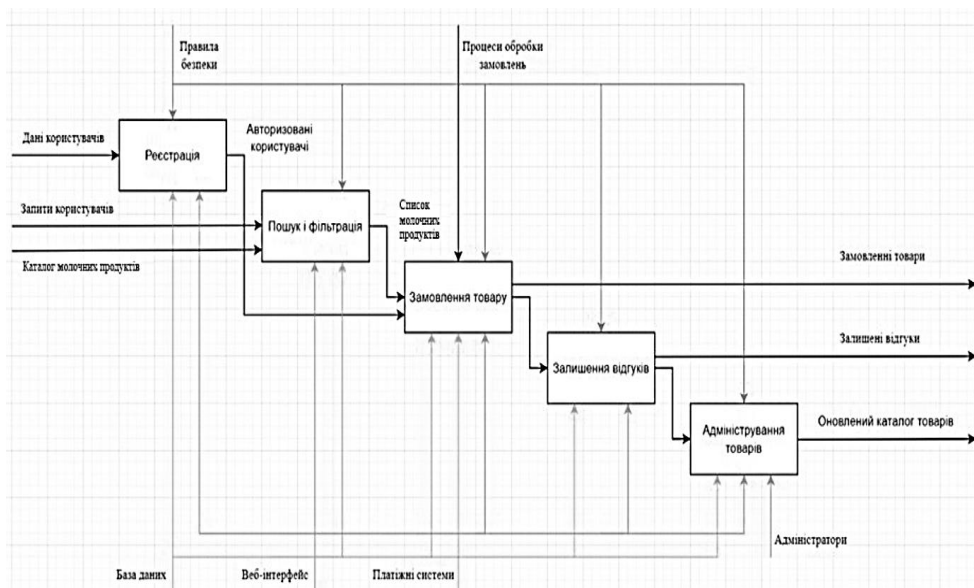


Рисунок 2.4 – Декомпозиція діаграми IDEF0

Діаграми компонентів відображають структурний взаємозв'язок між елементами програмної системи і допомагають зрозуміти, чи всі функціональні вимоги були враховані під час запланованої розробки.

Діаграми компонентів стають важливими для використання, коли ми проектуємо та будуємо складні системи.

Інтерфейси використовуються компонентами системи для спілкування один з одним.

В загальному випадку діаграма компонент складається із вихідних кодів та бінарних складових, а також може мати й виконані складові іншого типу й демонструє взаємозв'язки та взаємозалежності між різними компонентами програмного продукту.

Щоб змодельовати фізичні аспекти об'єктно-орієнтованих систем, які використовуються для специфікації, документування, візуалізації, побудови систем, що виконуються на основі прямої і зворотної інженерії програмного забезпечення якраз і доцільно застосовувати діаграми компонентів. Іншими словами діаграми компонентів можна ще представити як діаграми класів, що здійснюють свій фокус на компонентах певної системи і їх досить часто застосовують, щоб змодельовати статичну реалізацію системи.

У більшості випадків характеристикою клієнт-серверної архітектури загальна поведінка потоку даних є подібною для багатьох модулів у програмній системі. Загальна схема складається із надсилання клієнтом даних на сервер, очікуючи на потрібну відповідь, при цьому здійснюючи їх обробку за допомогою вже заданих перетворень або певних змін. А це означає, що користуватися вказаним ресурсом (вебзастосунком) можуть лише користувачі, які зареєструвались та в подальшому авторизувались. Щоб створювати замовлення користувачеві потрібно зареєструватись в даному застосунку. Для здійснення реєстрації необхідно зробити наступне:

– заповнення реєстраційної форми, шляхом введення логіну, паролю, електронної адреси, імені;

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		29

- перевірка валідності даних;
- збереження даних у таблицях та надсилання покупцеві повідомлення про вдале створення акаунту;
- введення покупцем даних для здійснення авторизації.

Відповідно до визначеної ролі, а саме зареєстрований користувач, незареєстрований користувач або адміністратор надається певний перелік дій, які може виконувати користувач (покупець). У випадку ролі адміністратора користувачі можуть модерувати контент сайту, відгуки, добавляти та видаляти товар тощо.

Рисунок 2.5 наочно демонструє за допомогою діаграми UML як здійснюється авторизація користувача. Прецеденти, що подаються із загальними описами надають розробникам необхідну та точну інформацію про всі програмні модулі, а саме про їх функціональні можливості, що спрощує значно роботу, оскільки немає потреби створювати нові модулі.

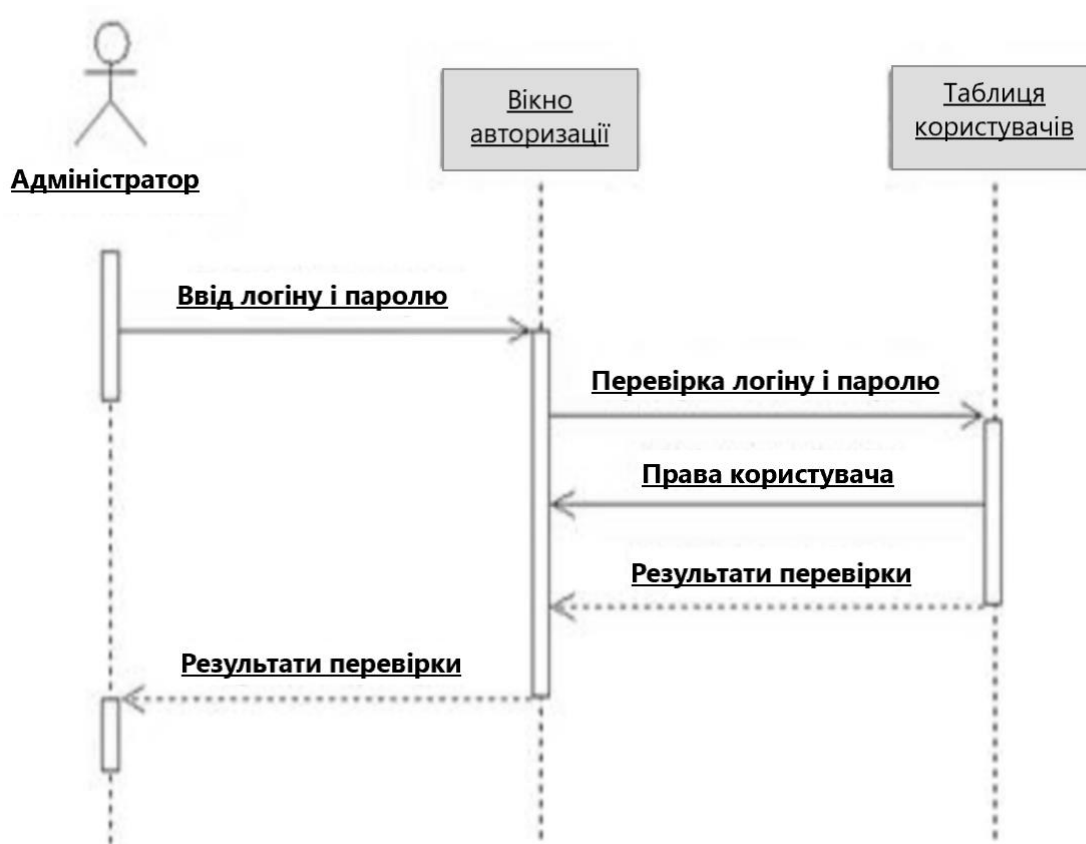


Рисунок 2.5 – Діаграма послідовності «Авторизація»

На початку перевіряється чи правильно введені дані. Якщо введені дані вірні, то відбувається перевірка прав доступу до функціоналу вебзастосунку на основі визначених ролей. Якщо велись неправильні дані то валідація не відбудеться і буде видано повідомлення про це, наприклад, «Невірне введення даних», «Спробуйте ще раз».

Покроковий опис подано нижче:

1. На початку здійснюється введення користувачем тих даних, що були запропоновані ним під час реєстрації.

2. Після цього сервер отримує дані із подальшою обробкою та перевіркою на валідність (правильність).

3. Якщо введені дані неправильні, то етап валідації не проходить, тоді відправляється відповідне повідомлення сервером про некоректність та неможливість увійти на ресурс.

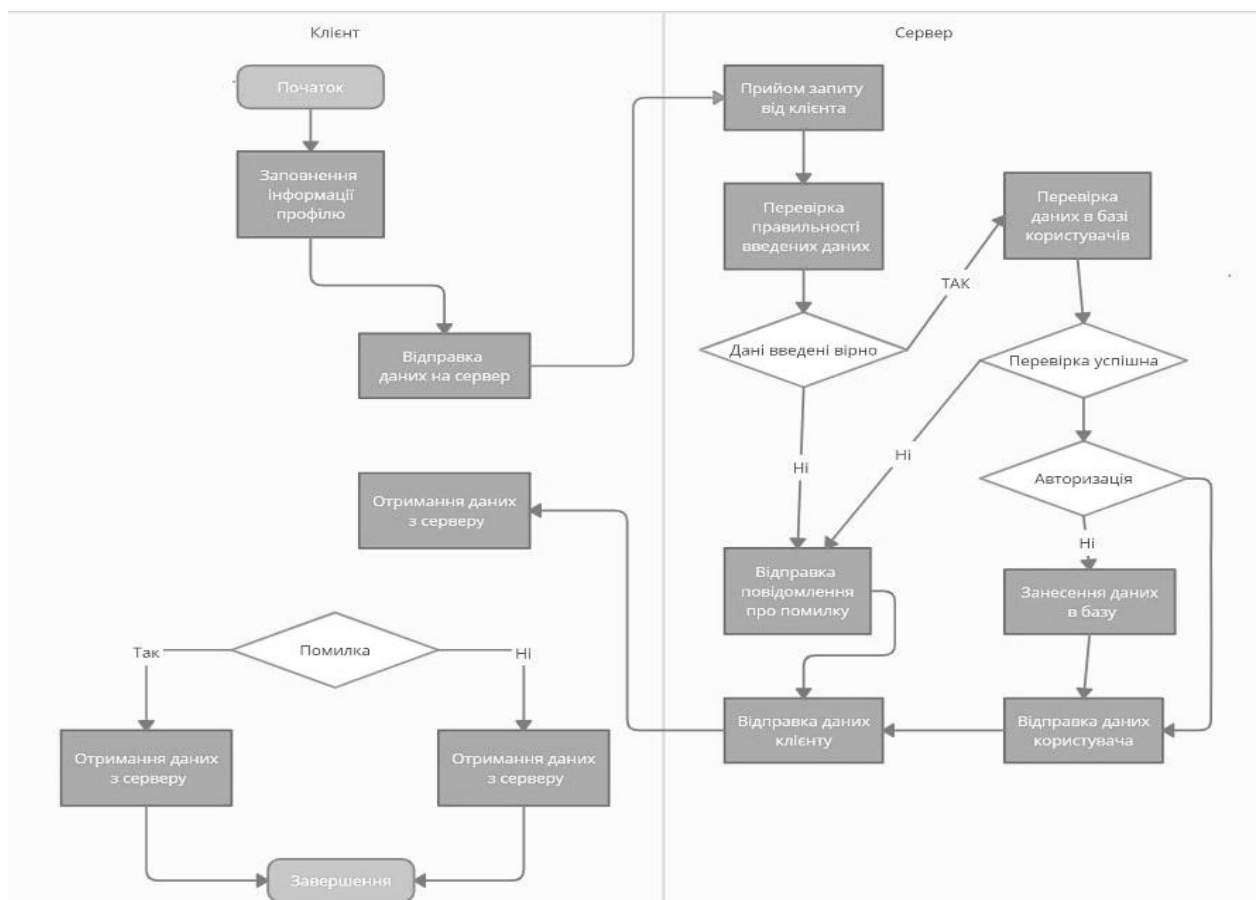


Рисунок 2.6 – Схема редагування профілю

Іншим процесом є «Заміна інформації користувача», який визначає дії, які відбуваються, коли користувач змінює особисту інформацію у своєму профілі. Покроковий процес редагування даних виглядає наступним чином. Спочатку користувач вводить нові дані профілю у відповідні поля та надсилає їх на сервер для перевірки та збереження. Далі сервер отримує дані та оновлює дані в профілі користувача після збереження. Після цього сервер починає розпізнавати профіль і перевіряє логін і пароль для зміни профілю, щоб запобігти редагування даних в іншому обліковому записі.

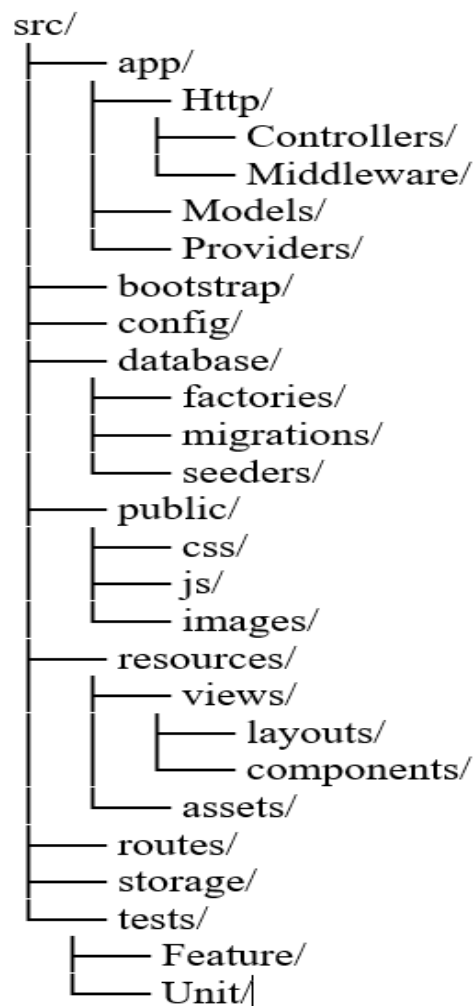


Рисунок 2.7 – Загальна структура вебзастосунку

## 2.3 Аналіз та вибір технологій і методів реалізації програмного забезпечення

З точки зору розробника, актуальні технології для веброзробки сьогодні представляють собою динамічний і постійно еволюціонуючий ландшафт. Далі розглянуто основні напрямки та конкретні технології, на які варто звернути увагу:

Для розробки клієнтської частини або фронтенду використовуються такі складові:

- основні мови та стандарти;
- фреймворки та бібліотеки для SPA (односторінкові вебзастосунки);
- інструменти для збірки та управління проектами.

До основних мов та стандартів відносяться HTML5, CSS3, JavaScript (ES6+). HTML5 є фундаментом для структури вебсторінок, постійно розвивається з новими семантичними тегами та API. CSS3, що відповідає за стилізацію та візуальне оформлення, включаючи потужні можливості для анімацій, гнучких макетів (Flexbox, Grid) та адаптивного дизайну (Media Queries). JavaScript (ES6+) - сучасні стандарти ECMAScript пропонують покращений синтаксис, нові можливості (async/await, класи, модулі) та роблять код більш читабельним та підтримуваним.

В якості фреймворків та бібліотек для SPA (Single Page Applications) використовують React, Angular, Vue.js, Svelte. React - лідируюча бібліотека для побудови складних інтерфейсів користувача на основі компонентного підходу та віртуального DOM. Екосистема React багата на інструменти для керування станом (Redux, Zustand, Context API), маршрутизації (React Router), тестування (Jest, React Testing Library) тощо. Angular - повноцінний фреймворк від Google, що пропонує структурований підхід до розробки великих вебзастосунків з використанням TypeScript. Має вбудовані рішення для керування станом (NgRx), маршрутизації, форм, HTTP-запитів та тестування. Vue.js -

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		33

прогресивний фреймворк, який легко інтегрується в існуючі проекти та підходить для розробки як простих інтерактивних елементів, так і складних SPA. Має зрозумілу документацію та активну спільноту. Для керування станом часто використовують Vuex або Pinia, для маршрутизації - Vue Router. Svelte - компілює код у невеликі, нативні JavaScript-модулі, що призводить до високої продуктивності та меншого розміру бандла. Пропонує свіжий підхід до розробки інтерфейсів.

Інструменти для збірки та управління проектами: Webpack, Parcel, Vite, npm/Yarn/pnpm. Webpack - потужний бандлер модулів, що дозволяє оптимізувати ресурси, транспілювати код (Babel), мінімізувати файли тощо. Parcel - Zero-configuration бандлер, який швидко налаштовується та підходить для швидкої розробки. Vite - швидкий інструмент для розробки фронтенду наступного покоління, що використовує нативний ESM для розробки та Rollup для збірки. npm/Yarn/pnpm/apt - менеджери пакетів для управління залежностями проекту.

CSS-препроцесори та методології: Sass/SCSS, Less розширюють можливості CSS, додаючи змінні, міксини, успадкування тощо; CSS Modules ізолюють стилі компонентів, запобігаючи конфліктам імен класів; Styled Components/Emotion дозволяють писати CSS безпосередньо в JavaScript-компонентах; Tailwind CSS/Bootstrap/Material UI - CSS-фреймворки та бібліотеки компонентів, що пропонують готовий набір стилів та елементів інтерфейсу для швидкої розробки.

Для розробки серверної частини застосовують такі засоби та середовища:

- мови програмування;
- бази даних;
- контейнеризація та оркестрація;
- хмарні технології.

Для створення бекенду або серверної частини використовуються такі мови програмування як JavaScript, що представляється фреймворком Node.js, Python

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		34

(Django, Flask), Java (Spring), PHP (Laravel, Symfony), Go (Gin, Echo), C# (.NET Core), Ruby (Ruby on Rails).

JavaScript (Node.js), що дозволяє використовувати JavaScript як на фронтенді, так і на бекенді, забезпечуючи єдину екосистему розробки. Популярний для створення API, реального часу та масштабованих застосунків. Python (Django, Flask) - універсальна мова з великою кількістю фреймворків для швидкої розробки вебзастосунків різної складності. Django - повноцінний фреймворк, Flask - мікрофреймворк. Java (Spring) - потужна та масштабована платформа для розробки корпоративних застосунків. Spring Framework пропонує широкий спектр інструментів та концепцій. PHP (Laravel, Symfony) - залишається популярною мовою для веб-розробки, особливо з такими сучасними фреймворками, як Laravel та Symfony, що пропонують зручний синтаксис та розвинену екосистему. Go (Gin, Echo) - компільована мова, відома своєю продуктивністю та паралелізмом, часто використовується для створення мікросервісів та високопродуктивних API. C# (.NET Core) - потужна платформа від Microsoft для розробки різноманітних застосунків, включаючи веб. .NET Core є кросплатформним. Ruby (Ruby on Rails) це фреймворк, що пропагує конвенції та швидку розробку.

В якості баз даних використовуються реляційні бази даних, графові бази даних. Далі наведено приклади таких баз даних. Реляційні (SQL), наприклад, PostgreSQL, MySQL, MariaDB мають характеристики потужних та надійних систем для структурованих даних. NoSQL MongoDB, Couchbase, Cassandra відносяться до більш гнучкіших рішень для неструктурованих або слабоструктурованих даних, часто використовуються для масштабованих застосунків. Графові бази даних Neo4j використовуються для роботи з даними, що мають складні зв'язки.

Кешування відбувається за допомогою Redis, Memcached, що слугують для підвищення продуктивності шляхом кешування часто використовуваних даних.

					КВРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		35

В якості API (Application Programming Interface) використовується REST, GraphQL, gRPC. REST (Representational State Transfer) – архітектурний стиль для створення масштабованих та гнучких API на основі протоколу HTTP. GraphQL - Мова запитів для API, що дозволяє клієнтам отримувати лише необхідні дані, підвищуючи ефективність. gRPC - високопродуктивний фреймворк для віддалених процедурних викликів (RPC) від Google, що використовує Protocol Buffers для серіалізації даних.

Для контейнеризація та оркестрації використовуються такі засоби, як Docker, Kubernetes. Docker - платформа для створення, розгортання та запуску застосунків у контейнерах, забезпечуючи ізоляцію та відтворюваність середовища. Kubernetes - система оркестрації контейнерів для автоматизації розгортання, масштабування та управління контейнеризованими застосунками.

В якості хмарних технологій використовуються AWS, Serverless. (Amazon Web Services), Google Cloud Platform (GCP), Microsoft Azure - набір хмарних сервісів, що надають інфраструктуру, бази даних, сервіси для розробки та розгортання вебзастосунків. Serverless, наприклад, AWS Lambda, Google Cloud Functions, Azure Functions - дозволяють запускати бекенд-код у відповідь на події без необхідності керування серверами.

Для розробки даного програмного продукту було обрано такі технології.

Бекенд:

- PHP 8.2.28;
- Laravel 12.12.0;
- Composer 2.5;
- База даних PostgreSQL 14.

Фронтенд:

- HTML5;
- CSS3;
- JavaScript (Vanilla JS);
- Livewire: для інтерактивних компонентів.

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		36

Середовище розробки:

- для локальної розробки і деплою використано Docker;
- вебсервер Nginx;
- хостинг платформа для деплойменту Render.com.

## 2.4 Проектування інтерфейсу користувача

Беручи до уваги функціональні та нефункціональні вимоги, ось підхід до проектування користувацького інтерфейсу (UI) для вебзастосунку для реалізації молочної продукції:

Загальні принципи дизайну вебзастосунків:

- простота і зручність використання;
- візуальна привабливість;
- оперативність;
- доступність;
- брендинг.

Інтерфейс повинен бути інтуїтивно зрозумілим, що дозволить користувачам легко знаходити продукти та здійснювати покупки. Потрібно використовувати якісні зображення молочних продуктів, щоб привабити користувачів, а також забезпечити безперебійну роботу на різних пристроях (настільних комп'ютерах, планшетах, смартфонах). Доцільно дотримуватись рекомендацій щодо доступності (WCAG) аби забезпечити зручність для людей з обмеженими можливостями. Важливим моментом є відображення ідентичності бренду (логотип, кольори, шрифти) у всьому дизайні.

До ключових компонентів інтерфейсу відноситься (прототип головної сторінки показано на рисунку 2.8):

- головна сторінка;
- банер з акціями або новими продуктами;
- популярні категорії молочних продуктів;

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		37

- рекомендовані продукти;
- відгуки клієнтів;
- новини або розділ блогу (за наявності).

Нижній колонтитул має містити контактну інформацію, посилання на соціальні мережі та інформацію про авторські права.

Сторінка каталогу має містити:

- список продуктів із зображеннями, назвами, цінами та кнопками «Додати в кошик»;
- фільтри для категорій, брендів, поживної цінності та алергенів;
- опції сортування (наприклад, за ціною, популярністю);
- посторінкову або нескінченну прокрутку для навігації по великому каталогу.

Сторінка продукту має містити:

- великі, якісні зображення продукту;
- детальний опис продукту з інформацією про поживну цінність та інгредієнти;
- інформацію про ціну та наявність;
- поле для введення кількості;
- кнопку «Додати в кошик».

Загалом для кращого розуміння, що повинен містити інтерфейс використовується прототипування. Також, загальновідомо, що зворотній зв'язок з кінцевим користувачем є дуже важливим, тому експерти пропонують використовувати прототипування для покращення цього зворотного зв'язку. Прототип є «швидким», оскільки він може бути автоматично згенерований на основі специфікації вимог.

Швидке прототипування намагається поєднати два аспекти поведінковий та графічний, що пов'язані між собою, але зберігаються окремо. Це дозволяє аналітику зосередитися лише на поведінці системи на початковому етапі, не відволікаючись на деталі користувацького інтерфейсу.

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		38



Рисунок 2.8 – Прототип Головної сторінки вебзастосунку

Рисунок 2.8 демонструє прототип Головної сторінки інтерфейсу вебзастосунку для торгівлі молочними продуктами. Як бачимо з даного мокапу дизайн інтерфейсу має бути досить мінімалістичного стилю, містити невелику кількість кнопок та не перевантажувати візуальне сприйняття та не бути важким у користуванні.

Головна сторінка передбачає наявність логотипу фірми, поля Пошук, завдяки якому можна здійснити пошук певного виду молочної продукції. Також має бути передбачено можливість здійснити реєстрацію одразу з головної сторінки. Кнопка Меню допомагає обирати, що може робити користувач на даному сайті.

Також передбачена деталізація за кожною категорією і рекомендована продукція. Завдяки фільтрам можна здійснити пошук товарів за певними критеріями. На рисунку 2.9 продемонстровано прототип сторінки з фільтруванням за категоріями молочної продукції.

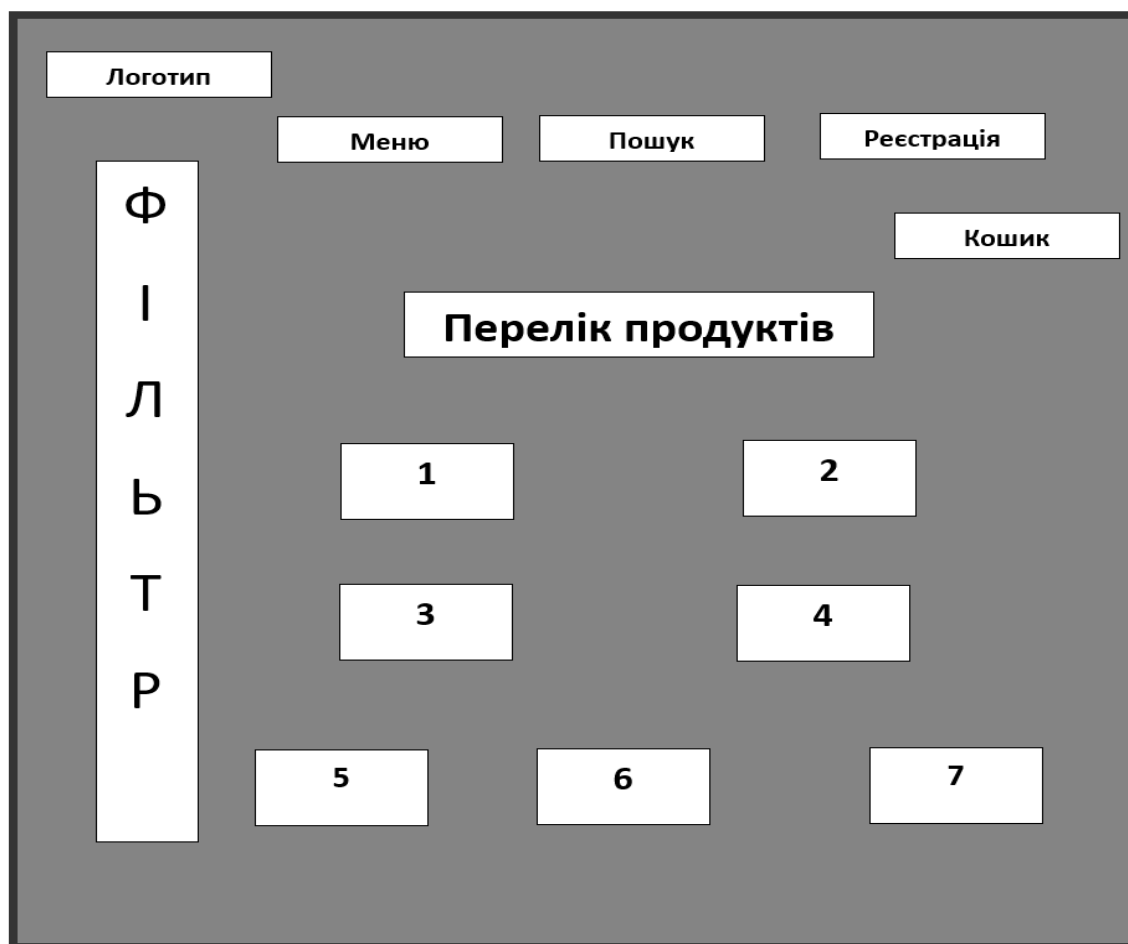


Рисунок 2.9 – Вигляд сторінки з фільтром

Сторінка із фільтром містить такі компоненти як Логотип, поле Пошук, Можливість реєстрації, кнопку Кошика. Також дане вікно має передбачати наявність переліку молочної продукції та фільтрування. Панель з можливістю фільтрування має розміщуватись з лівого боку. Зайшовши на цю панель, користувач матиме можливість обирати категорію товару обираючи діапазон цін, смакових характеристик, виробника, терміну споживання, харчових властивостей, загального опису товару тощо.

Також важливою сторінкою є сторінка із описом кожного конкретного молочного продукту. Прототип такого вікна зображено на рисунку 2.10.

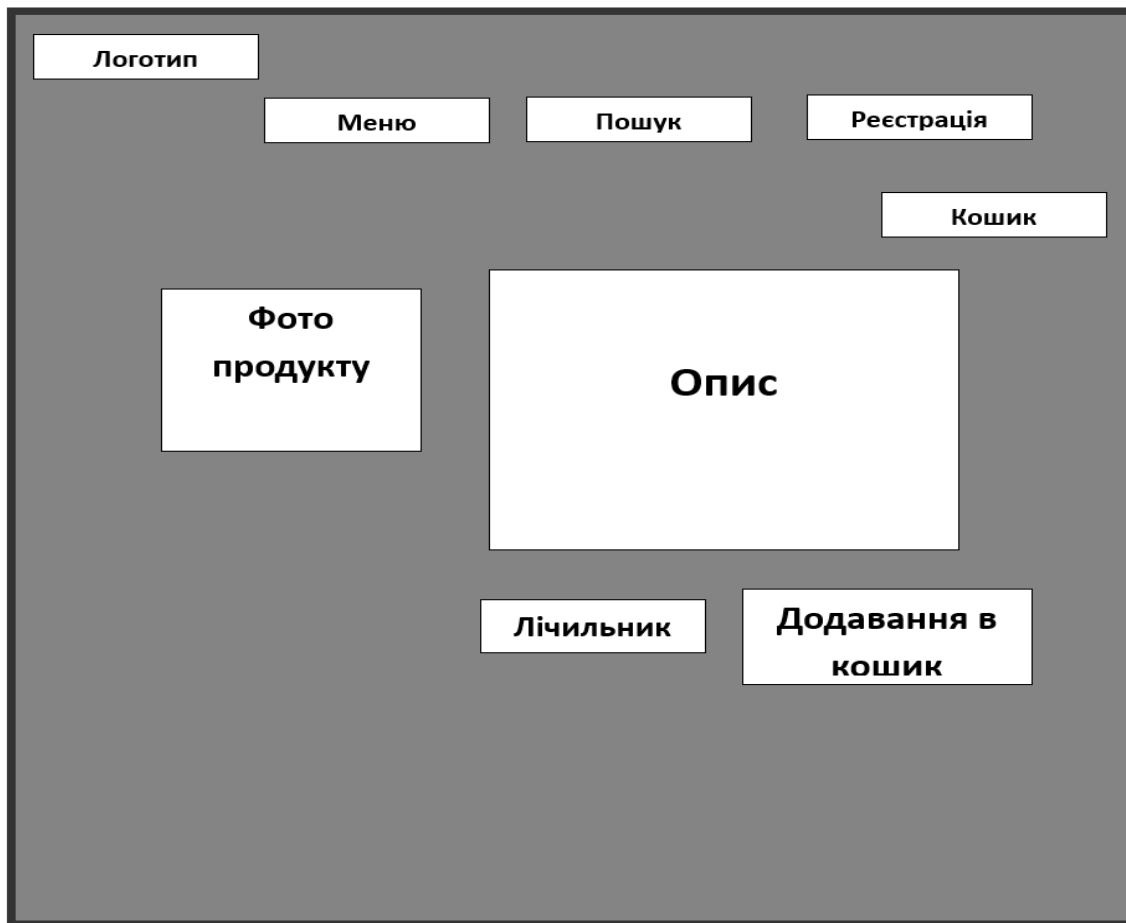


Рисунок 2.10 – Прототип вікна з описом продукції

Отже, як бачимо прототипування значно полегшує та спрощує процес візуалізації інтерфейсу користувача та допомагає при розробці якісного програмного забезпечення.

### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ

#### 3.1 Реалізація бази даних

Для успішної реалізації необхідно здійснити реалізацію бази даних. Створення надійної бази даних для вебзастосунків вимагає ретельного врахування кількох важливих факторів, зокрема структури даних, масштабованості, продуктивності, безпеки та цілісності даних. Добре структурована база даних забезпечує ефективне зберігання, пошук і маніпулювання даними для підтримки функціональності та зручності користування вебзастосунком. Загальна структура бази даних, що використовується у даному вебзастосунку представлено на рисунку 3.1.

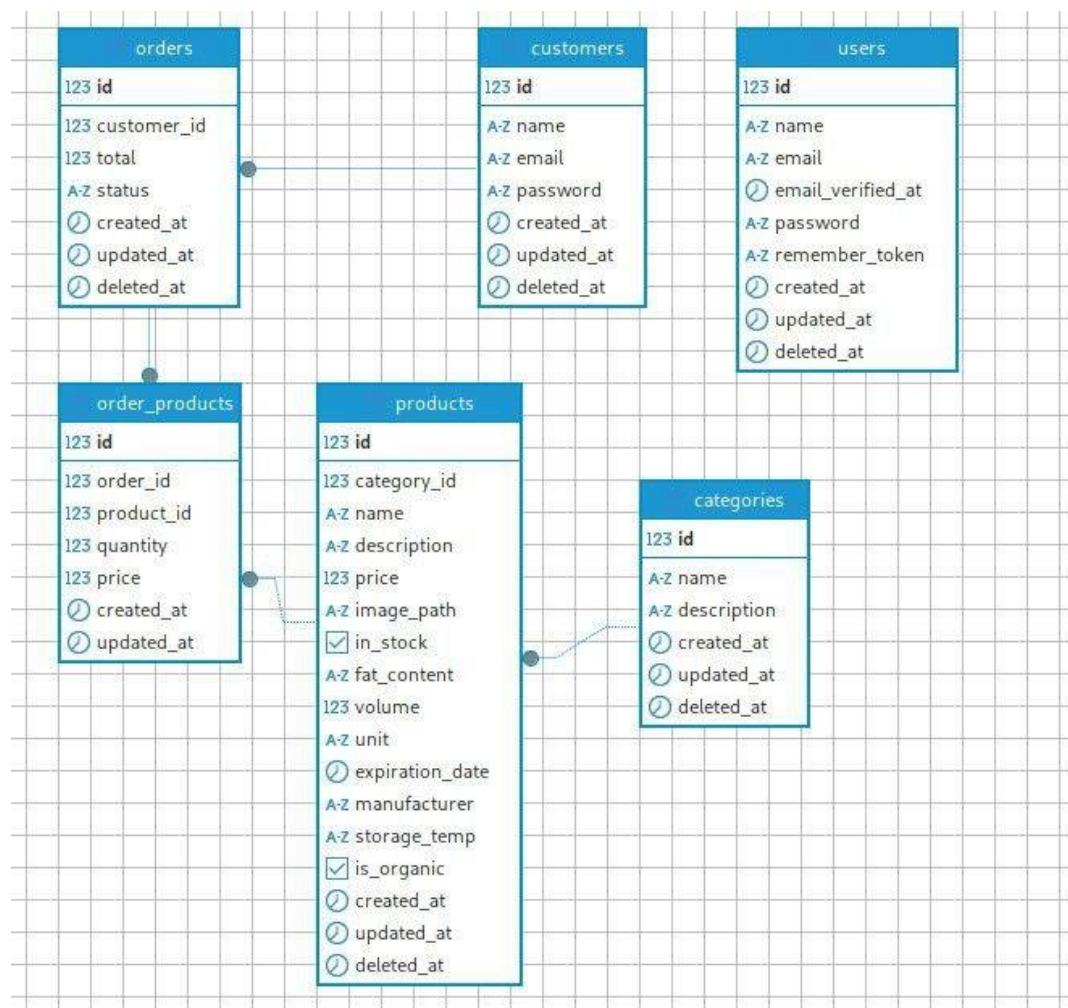


Рисунок 3.1 – Структура бази даних

Основні таблиці, з яких складається база даних: Categories, Products, Customers, Orders.

Таблиця 1 – Категорії (Categories)

id	унікальний ідентифікатор
name	назва категорії (унікальна)
description	опис категорії
created_at, updated_at	часові мітки

Таблиця 2 – Продукція (Products)

id	унікальний ідентифікатор
category_id	зовнішній ключ до таблиці categories
name	назва продукту
description	опис продукту
price	ціна
image_path	шлях до зображення
in_stock	наявність товару (boolean)
fat_content	вміст жиру
volume	об'єм
unit	одиниця виміру (ml, g)
expiration_date	термін придатності
manufacturer	виробник
storage_temp	температура зберігання
is_organic	органічний продукт (boolean)
created_at, updated_at	часові мітки
deleted_at	для soft deletes

Таблиця 3 – Покупці (Customers)

id	унікальний ідентифікатор
name	ім'я клієнта
email	електронна пошта (унікальна)
password	хешований пароль
phone	номер телефону
address	адреса
created_at, updated_at	часові мітки

Таблиця 4 - Заовлення (Orders)

id	унікальний ідентифікатор
customer_id	зовнішній ключ до таблиці customers
total	загальна сума заовлення
status	статус заовлення
shipping_address	адреса доставки
payment_method	спосіб оплати
created_at, updated_at -	часові мітки
order_items	заовлення
id	унікальний ідентифікатор
order_id	зовнішній ключ до таблиці orders
product_id	зовнішній ключ до таблиці products
quantity	кількість товару
price	ціна на момент заовлення
created_at, updated_at	часові мітки

### 3.2 Реалізація модулів вебзастосунку

Адміністративна панель реалізована як окремий модуль (filament:3.2) з контролем доступу на основі ролей.

Реалізовано наступний функціонал: управління продуктами, керування категоріями, управління користувачами, замовлення, аналітика, доступ до адміністративної панелі. Далі подано деталізований опис даного функціоналу.

Управління продуктами:

- додавання, редагування, видалення продуктів;
- керування атрибутами: ціна, опис, категорія, наявність;
- завантаження зображень.

Керування категоріями:

- створення та редагування категорій;
- встановлення ієрархії категорій.

Управління користувачами:

- перегляд списку зареєстрованих користувачів;
- редагування інформації та статусу користувачів.

Замовлення:

- перегляд та управління замовленнями;
- зміна статусу замовлень;
- експорт даних обраних замовлень.

Аналітика:

- основні показники (продажі, кількість замовлень);
- статистика популярності продуктів.

Доступ до адміністративної панелі:

- через URL /admin;
- потрібна аутентифікація з адміністраторськими правами;
- система логування дій адміністраторів.

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		45

Фронтенд реалізований на основі власних компонентів без використання великих JavaScript-фреймворків для збільшення швидкості завантаження із використанням таких технологій:

- Blade - шаблонізатор Laravel для відображення сторінок;
- Vanilla JavaScript - для інтерактивної взаємодії;
- CSS - власні стилі без використання фреймворків;
- Livewire - для AJAX-взаємодії без написання окремого JS.

Ключові компоненти інтерфейсу:

- головна сторінка з презентацією товарів;
- каталог продуктів з фільтрацією;
- сторінка пошуку за назвою і додатковими фільтрами та сортуванням;
- сторінка продукту з детальним описом;
- кошик для збереження обраних продуктів;
- форми реєстрації/входу для користувачів;
- кабінет кастомера.

Особливості:

- адаптивний дизайн для мобільних пристроїв;
- оптимізовані зображення для швидкого завантаження;
- модальні вікна для швидкої взаємодії (кошик, авторизація);
- асинхронне оновлення кошика без перезавантаження сторінки.

### 3.3 Реалізація інтерфейсу та інструкція користувача

Коли користувач перший раз відвідує розроблений ресурс, то він має зареєструватись, а в подальшому для коректної роботи, збереження всієї інформації про свій вибір товару, покупки, повідомлення, обговорення тощо необхідно буде авторизуватись.

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		46

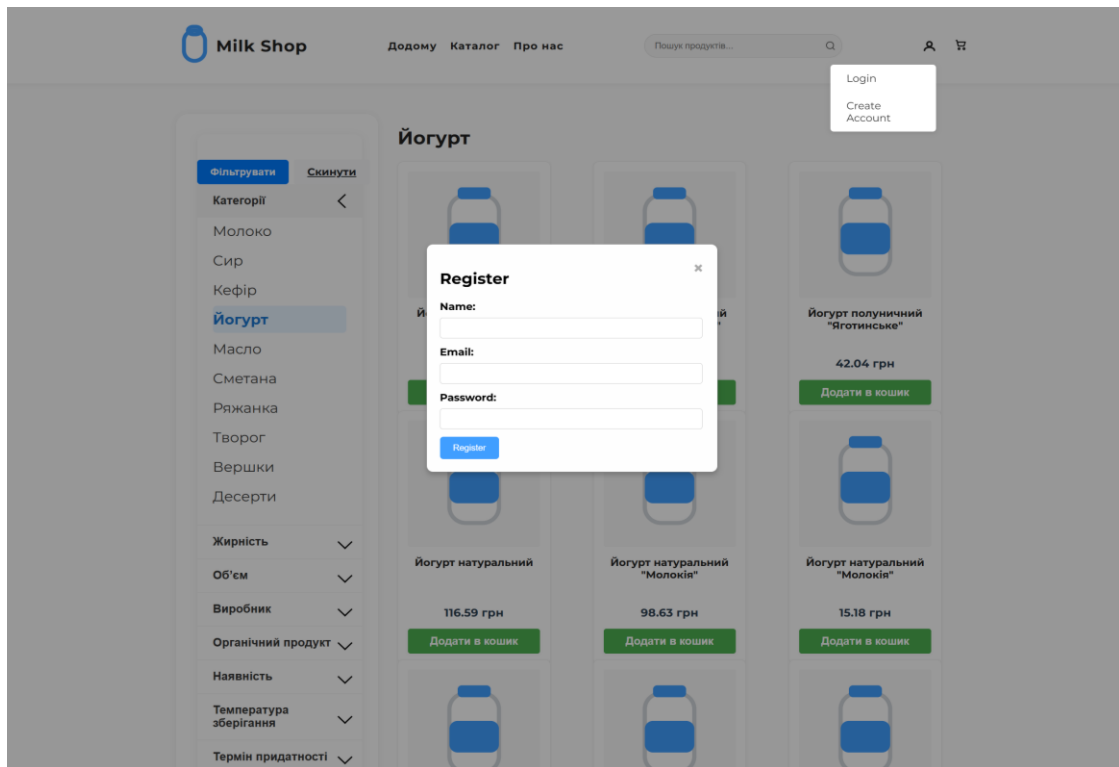


Рисунок 3.2 – Вікно для здійснення реєстрації користувача

Щоб здійснити реєстрацію на даному ресурсі необхідно натиснути на кнопку «Create Account». Після цього відкриється діалогове вікно, де потрібно здійснити введення імені у полі Name, електронну адресу у полі E-mail, придумати надійний пароль та ввести його у поле Password. Після здійснення всіх маніпуляцій потрібно натиснути на кнопку Register.

Якщо все зроблено та введено вірно, то користувач автоматично перейде на сторінку для здійснення авторизації та побачить відповідне повідомлення про успішність реєстрації на ресурсі та створенні акаунту.

Як відмічалось вище для успішного користування даним ресурсом необхідно в подальшому здійснювати авторизацію. Діалогове вікно для авторизації показано на рисунку 3.3.

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		47

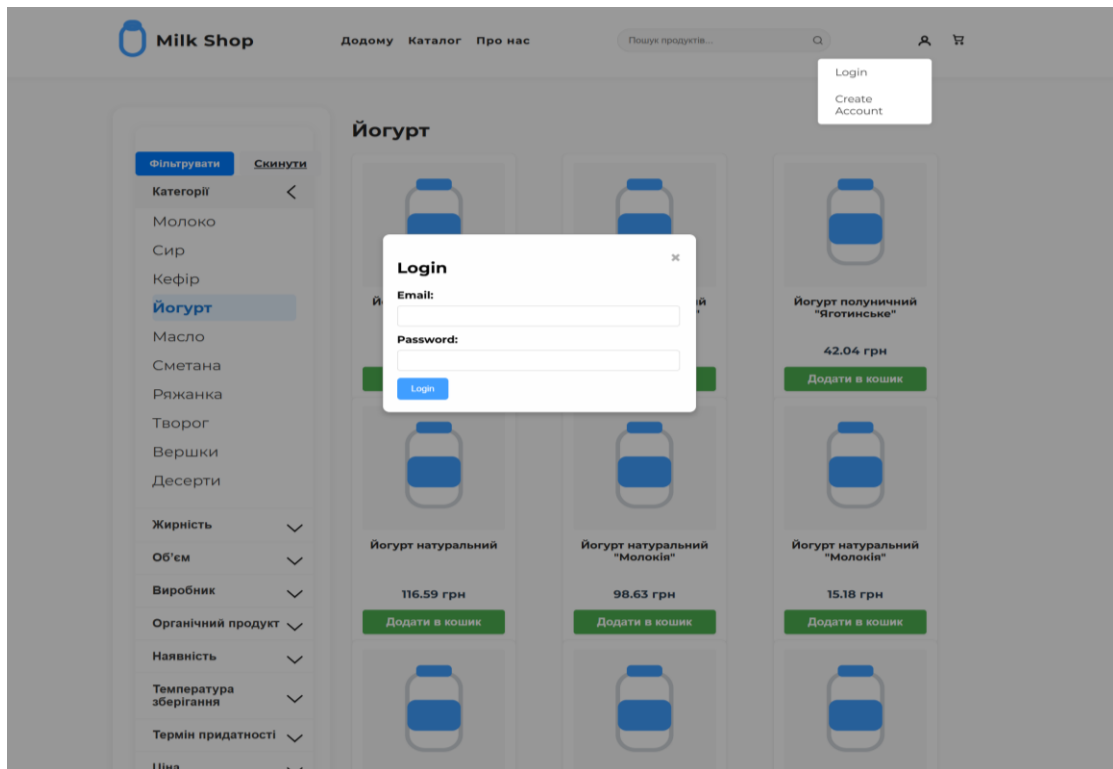


Рисунок 3.3 – Здійснення авторизації зареєстрованим користувачем

Щоб увійти на ресурс та успішно користуватись даним вебзастосунком потрібно ввести електронну адресу та пароль. Тобто у поле Email необхідно ввести електронну адресу, а у поле Password ввести пароль, зазначений під час реєстрації.

Якщо користувач або потенційний клієнт здійснив вдало авторизації, то він може повноправно користуватись даним вебзастосунком, а саме обирати молочну продукції, користуватись фільтром у каталозі, щоб мати розуміння про якість та властивості, а також характеристики того чи іншого молочного продукту.

Загалом вебзастосунок передбачає використання мінімалістичного дизайну інтерфейсу користувача. Меню складається з малої кількості пунктів: Додому, Каталог, Про нас. Натиснувши, наприклад, на пункт меню Про нас користувач може ознайомитись з інформацією щодо діяльності компанії, яка займається продажем молочної продукції. Коли натиснути на пункт меню Додому, то користувач перенаправиться на Головну сторінку, де буде

можливість здійснювати авторизацію, реєстрацію, пошук товару за фільтрами тощо.

Як уже зазначалось, якщо користувач зареєструвався та авторизувався, то він може обирати продукцію. На рисунку 3.4 подано вигляд сторінки із категоріями, де відображаються основні категорії молочної продукції, а саме такі вкладки: Молоко, Сир, Кефір, Йогурт, Масло, Сметана, Ряжанка, Творог, Вершки, Десерти.

Для прикладу, після натиснення на вкладку, наприклад Творог користувач перейде на сторінку вибору товару за категорією Творог. Відповідно користувач зможе обирати собі продукт, який його цікавить та добавляти його у кошик.

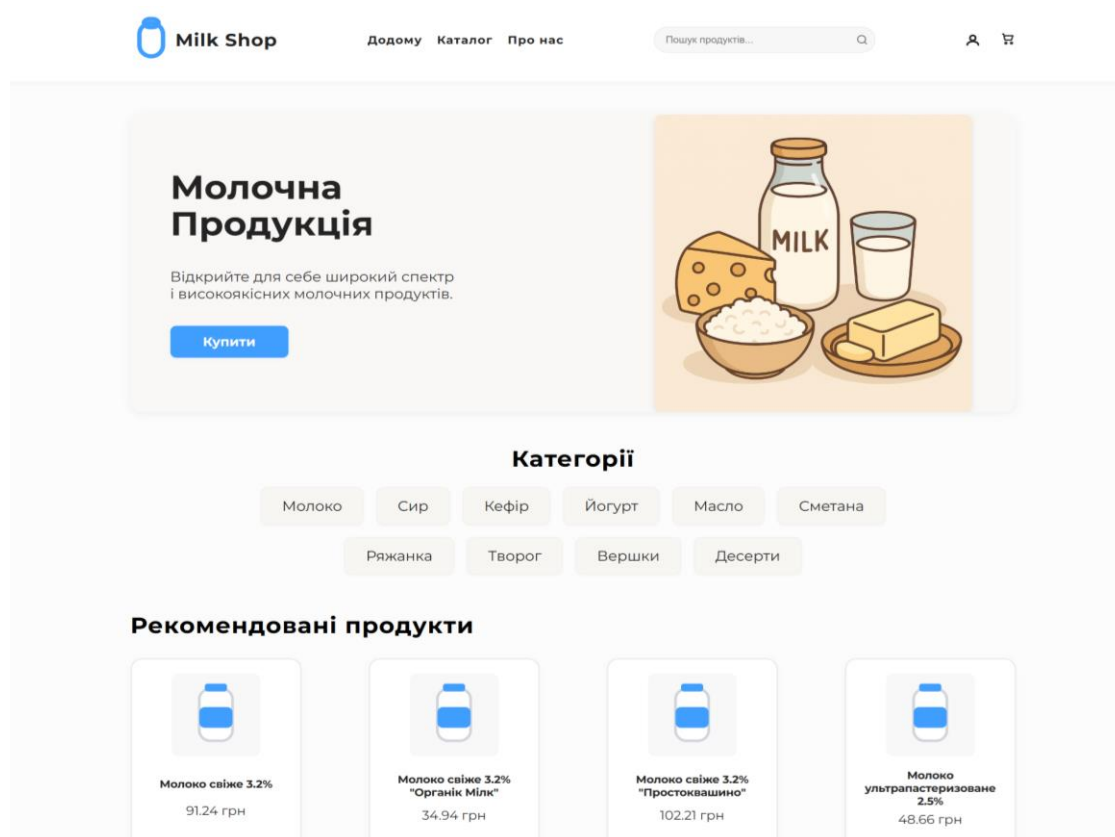


Рисунок 3.4 – Вигляд сторінки із категоріями

Також у відкритому вікні можна буде побачити таку функцію як рекомендовані товари.

					КВРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		49

Відповідно до обраної категорії покупець перейде на ту сторінку із молочною продукцією, яку він обрав. Тобто буде можливість користування фільтрами. Вікно із зображенням можливості використання фільтрів продемонстровано на рисунку 3.5.

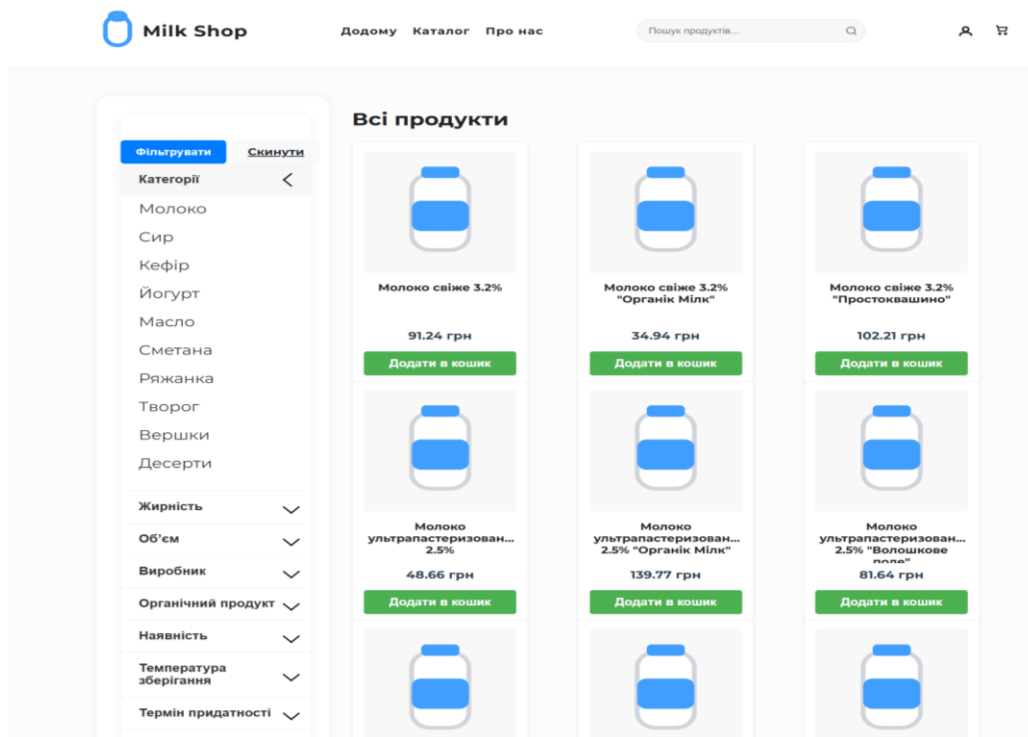


Рисунок 3.5 – Використання фільтру

Функціонал вебзастосунку дозволяє користувачеві детально ознайомитися з продукцією: для цього потрібно натиснути кнопку «Детальніше» у списку доступних товарів. Додавання позицій до кошика здійснюється кнопкою «Додати до кошика».

Переглянути вміст кошика та відредагувати його (наприклад, видалити помилково обраний товар) можна, перейшовши на сторінку кошика. На Рисунок 3.6 подано ілюстрацію сторінки, на якій можна здійснити вибір молочної продукції, перенести його до кошика і відповідно здійснити оплату та покупуц того товару, що цікавить покупця.

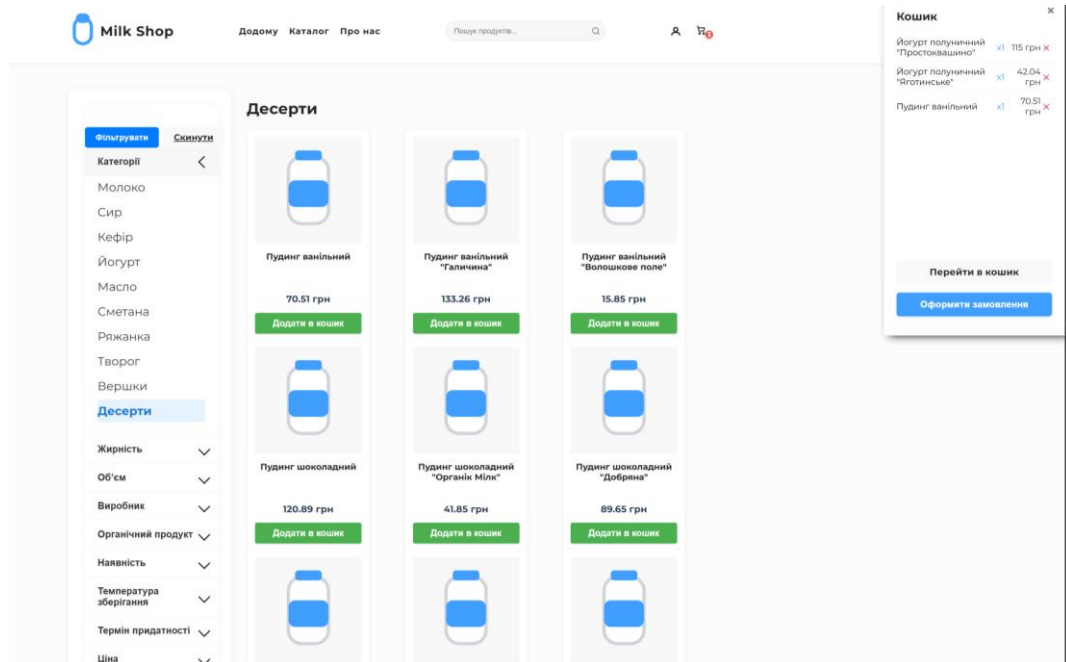


Рисунок 3.6 – Сторінка Кошик

На рисунку 3.6 продемонстровано сторінку вкладки Кошик, де відображається абсолютно вся інформація про той товар, що вибрав покупець: ціна, кількість, спосіб оплати, можливість доставки тощо.

Користуватися цим ресурсом інтуїтивно зрозуміло, адже його функціонал відповідає звичним правилам роботи з онлайн-платформами. Для початку роботи достатньо перейти за відповідною веб-адресою. Щоб отримати доступ до всіх можливостей сайту, користувачу необхідно пройти реєстрацію, ввівши свої дані, логін та пароль, а потім здійснити авторизацію.

Після успішного входу, користувач може легко обирати потрібні категорії товарів, а потім і самі товари. Вибір продукції здійснюється шляхом додавання її до кошика, де можна переглянути обрані позиції, видалити зайві та перейти до оплати за допомогою спеціальної кнопки.

Після натискання на кнопку Оформити замовлення відбудеться автоматичне перенаправлення на діалогове вікно із можливістю оформлення замовлення, вибору способу оплати, відмітки про зворотній зв'язок та підтвердження замовлення, вибору способу доставки.

### 3.4 Тестування вебзастосунку

Тестування програмного забезпечення є критично важливим елементом забезпечення якості програмного забезпечення і являє собою остаточну перевірку специфікації, дизайну, кодування.

Цілі тестування:

- тестування - це процес виконання програми з метою пошуку помилки;
- хороший тестовий приклад - це той, який має високу переносимість пошуку ще не виявленої помилки;
- успішний тест - це тест, який виявляє ще не виявлену помилку.

Основним завданням тут буде розробка тестових кейсів для виявлення різних класів помилок і зробити це з мінімальними витратами часу і зусиль. Якщо тестування буде проведено успішно, воно виявить помилки в програмному забезпеченні. Ще однією перевагою є те, що воно демонструє, що функції програмного забезпечення працюють відповідно до специфікацій і вимоги до продуктивності були виконані.

Але тестування не може показати відсутність дефектів, воно може показати лише наявність помилок у програмному забезпеченні.

Тестування - це процес вивчення продуктів для визначення того, які дефекти вони містять, щоб провести тест, програмісти повинні мати вже створене програмне забезпечення, а також повинен існувати чітко визначений стандарт для того, що є дефектом. Розробники можуть тестувати продукти, переглядаючи їх конструкцію та склад, або виконуючи їх функцію та досліджуючи результат.

Інформаційна система - це інтегрований набір програмних компонентів. Компоненти можна тестувати окремо або в групах, а можна тестувати всю систему в цілому. Тестування компонентів окремо називається «інтеграційним тестуванням». Тестування всієї системи називається «системним тестуванням». Тестування модуля називається «модульне тестування».

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		52

Ці три типи тестування співвідносяться з певною фазою життєвого циклу розробки системи (SDLC - System Development Life Cycle).

Системне тестування перевіряє поведінку всієї системи з точки зору вимог технічного користувача. Ці вимоги визначаються на етапі аналізу життєвого циклу розробки системи. Інтеграційне тестування перевіряє поведінку різних груп програмних компонентів. Модульне тестування тестує кожен окремий програмний компонент ізольовано.

Оскільки кожен рівень тестування пов'язаний з певною фазою SDLC, діяльність з тестування може бути розподілена по всьому життєвому циклу. Планування кожного типу тестування може відбуватися на відповідній фазі SDLC.

Тестові дані генеруються за допомогою Laravel Factories та Seeders.

Factory:

Фабрики знаходяться в директорії `database/factories/` і використовують бібліотеку Faker для генерації реалістичних даних:

`CategoryFactory.php` - створює тестові категорії

`ProductFactory.php` - створює тестові продукти

`CustomerFactory.php` - створює тестових користувачів

Seeder:

Сідери знаходяться в директорії `database/seeders/` і відповідають за заповнення бази даних тестовими даними:

`CategorySeeder.php` - створює базові категорії молочних продуктів

`ProductSeeder.php` - створює продукти з різними характеристиками

`AdminUserSeeder.php` - створює адміністраторів системи

`DatabaseSeeder.php` - головний сідер, який викликає всі інші

**Запуск фікстур:**

*# Розгортання структури бази даних*

`php artisan migrate`

*# Заповнення тестовими даними*

`php artisan db:seed`

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		53

# Специфічний сідер  
php artisan db:seed --class=ProductSeeder

```
milkshop-w9rt / Logs
Logs
All logs v Search
May 7 10:53:20 PM INFO ✓ homepage loads correctly 0.72s
May 7 10:53:20 PM INFO ✓ catalog page loads 0.42s
May 7 10:53:20 PM INFO ✓ about page loads 0.37s
May 7 10:53:21 PM INFO PASS Tests\Feature\CategoryPageTest
May 7 10:53:21 PM INFO ✓ category page displays all products 0.49s
May 7 10:53:21 PM INFO ✓ empty category page 0.35s
May 7 10:53:21 PM INFO ✓ nonexistent category returns 404 0.37s
May 7 10:53:23 PM INFO PASS Tests\Feature\HomePageTest
May 7 10:53:23 PM INFO ✓ home page loads successfully 0.41s
May 7 10:53:23 PM INFO ✓ home page displays products and categories 0.39s
May 7 10:53:23 PM INFO ✓ home page has search form 0.32s
May 7 10:53:24 PM INFO PASS Tests\Feature\ProductDetailsTest
May 7 10:53:24 PM INFO ✓ product details page displays correctly 0.60s
May 7 10:53:24 PM INFO ✓ nonexistent product returns 404 0.40s
May 7 10:53:24 PM INFO PASS Tests\Feature\ShoppingCartTest
May 7 10:53:24 PM INFO ✓ cart session storage works 0.20s
May 7 10:53:24 PM INFO PASS Tests\Feature\UserProfileTest
May 7 10:53:24 PM INFO ✓ customer can be created 0.12s
May 7 10:53:24 PM INFO ✓ customer password is hashed 0.17s
May 7 10:53:24 PM INFO Tests: 26 passed (69 assertions)
May 7 10:53:24 PM INFO Duration: 9.49s
May 7 10:53:24 PM INFO Some tests failed but continuing deployment as this is a non-blocking check
May 7 10:53:24 PM INFO Test results reported. Deployment continues regardless of test results.
May 7 10:53:24 PM INFO Publishing Livewire assets...
```

Рисунок 3.10 – Приклад виконання тестів при деплої

Проект підтримує декілька середовищ виконання з різними конфігураціями.

### Локальне середовище:

- Docker Compose для локальної розробки;
- файл конфігурації: .env;
- база даних: PostgreSQL в докер-контейнері;
- веб-сервер: Nginx в докер-контейнері.

### Запуск:

- docker-compose up -d.

### Середовище розробки:

- Render.com для хостингу

- файл конфігурації: .env.render;
- база даних: Render PostgreSQL Service;
- запуск: автоматичний при пуші в репозиторій.

```

Logs
All logs Search
May 7 10:36:31 PM INFO ✓ search products with price sorting 0.38s
May 7 10:36:31 PM INFO PASS Tests\Feature\ShoppingCartTest
May 7 10:36:31 PM INFO ✓ cart session storage works 0.12s
May 7 10:36:31 PM INFO PASS Tests\Feature\UserProfileTest
May 7 10:36:31 PM INFO ✓ customer can be created 0.18s
May 7 10:36:31 PM INFO ✓ customer password is hashed 0.10s
May 7 10:36:31 PM INFO FAILED Tests\Feature\ProductSearchTest > search products by keyword
Expected: <!DOCTYPE html>\n
<html lang="uk">\n
<head>\n
... (672 more lines)
To contain: Молоко 3.2%
at tests/Feature/ProductSearchTest.php:78
74 | {
75 |     $response = $this->get('/search?query=молоко');
76 |
77 |     $response->assertStatus(200);
+ 78 |     $response->assertSee('Молоко 3.2%');
79 |     $response->assertSee('Молоко знежирене 0.5%');
80 |     $response->assertDontSee('Йогурт полунічний');
81 | }
82 |
Tests: 1 failed, 28 passed (75 assertions)
Duration: 9.61s

```

Рисунок 3.11 – Приклад виявленої помилки при виконанні тестів

## Розгортання на Render:

Налаштування файлу render.yaml:

- Web Service для PHP додатку;
- PostgreSQL для бази даних;
- налаштування env змінних;
- автоматичний деплой при пуші в main гілку.

Загалом на початку написання даної кваліфікаційної роботи було поставлено низку задач, після вирішення яких передбачалось проектування та

					КВРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		55

розробка вебзастосунку для торгівлі молочною продукцією. У таблиці 3.1 показано опис цих завдань та відмітка про виконання чи невиконання.

Таблиця 3.1 – Аналіз стану виконання завдання

<b>Завдання</b>	<b>Виконання</b>
Проаналізувати ринок реалізації молочної продукції в мережі Інтернет;	виконано
Провести аналітичний огляд існуючих програмно-технічних рішень, а саме вебресурсів з продажу молокопродуктів із виділенням негативних та позитивних практик реалізації;	виконано
Визначити функціональні та нефункціональні вимоги розроблюваного програмного застосунку;	виконано
Розробити технічне завдання;	виконано
Здійснити вибір типу архітектури та проектування бази даних;	виконано
Здійснити реалізацію програмного застосунку для реалізації молочної продукції;	виконано
Протестувати вебзастосунок для реалізації молочної продукції.	виконано

## ВИСНОВКИ

Отже, на основі здійсненої роботи над курсовим проектом можна зробити такі висновки.

У першому розділі було здійснено аналіз предметної області та виявлено, що клієнтоорієнтований дизайн є ключовим для успішного вебзастосунку. Важливо, щоб структура та логіка сайту були інтуїтивно зрозумілими та послідовними для покупців різних вікових та соціальних груп. Зокрема, ми наголошуємо на необхідності спростити навігацію та додати функцію розумного пошуку, щоб користувачі могли швидко знайти потрібний товар.

Загалом, розробка такого роду вебзастосунків, інтернет-магазинів та загалом вебресурсів різного роду сьогодні є дуже актуальною, оскільки дозволяє бізнесам переносити свою діяльність в онлайн-простір, що особливо важливо в умовах військових дій в Україні.

Другий розділ присвячений проектуванню архітектури програмного продукту. Тут ми детально описуємо його структурні компоненти, доповнюючи їх відповідними діаграмами. Також у цьому розділі обґрунтовується вибір середовища та інструментарію для розробки.

Для проекту обрано клієнт-серверну архітектуру. Це обчислювальна модель, де сервер є центральним сховищем, постачальником та менеджером більшості ресурсів і послуг, що споживаються клієнтами. У такій архітектурі один або кілька клієнтських комп'ютерів підключаються до центрального сервера через мережу або інтернет. Цю модель також називають мережевою обчислювальною моделлю або мережею клієнт-сервер, оскільки всі запити та послуги передаються через мережу.

У третьому розділі представлено реалізацію програмного продукту. Крім того, тут надається інструкція для користувача, а також викладено методику та результати тестування програмного забезпечення.

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		57

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Кваліфікаційна робота : Методичні настанови для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 121 «Інженерія програмного забезпечення» / Л. П. Бедратюк, Г. І. Радельчук. Хмельницький : ХНУ, 2023. 60 с.
2. Методи та засоби визначення показників якості продукції : навчальний посібник / Т.З. Бубела та ін. Львів : Львівська політехніка, 2012. 292 с.
3. Functional and Nonfunctional Requirements: Specification and Types. URL: <https://www.altexsoft.com/blog/functional-and-non-functional-requirements-specification-and-types/> (дата звернення 15.10.24)
4. Види бази даних. URL: <https://futurenow.com.ua/shho-take-bazy-danyh-yih-pryznachennya-ta-vydy/> (дата звернення 22.11.24).
5. Інтернет-магазин з продажу молочної продукції «Галичина» URL: <https://dostavka.galychyna.com.ua/> (дата звернення 08.04.2025).
6. Інтернет-магазин з продажу молочної продукції «Sladosvit» URL: <https://sladosvit.com/> (дата звернення 08.04.2025).
7. Найбільший оптовий інтернет-ринок сільськогосподарської продукції в західній регіоні України URL: <https://vip.shuvar.com/fermers-ki-produkti/domashni-molochni-produkti-ta-jajce> (дата звернення 08.04.2025).
8. Інтернет-магазин з продажу молочної продукції «Molokoferma» URL: <https://molokoferma.com.ua/product-category/molochni-produkty/> (дата звернення: 08.04.2025).
9. Інтернет-магазин з продажу молочної продукції. URL: <https://gogoliv.com/product-category/milk/> (дата звернення: 08.04.2025).
10. Why eCommerce sales are expected to grow. URL: <https://www.dairyfoods.com/articles/97655-why-ecommerce-sales-are-expected-to-grow> (дата звернення: 08.04.2025).

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		58

11. King A. J., Lazard A. J. & White S. R. The influence of visual complexity on initial user impressions: Testing the persuasive model of web design. *Behaviour & Information Technology*. № 39(5). P. 497–510.

12. Lee, M. S. A. & Singh, J. The landscape and gaps in open source fairness toolkits. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 699 (2021).

13. Ageed Z. S. *et al.* Comprehensive survey of big data mining approaches in cloud systems. *Qubahan Acad. J.* № 1(2). P. 29–38.

14. Kurer and Oskar. The Political foundations of economic development policies. *Journal of Development Studies*. 2018. №32. P. 645-668.

15. Nath D., Roy A. Average energy and Shannon entropy of a confined harmonic oscillator in a time-dependent moving boundary *Journal of Mathematical Chemistry*. 2023. №61. P. 1491-1503.

16. Ananjevskii S. M., Chen A. P. A continuous version of the selfish parking problem *Vestnik St Petersburg University Mathematics*. 2024. № 57. P. 53-61.

17. Best Practices for Web Application Development. URL: <https://naturally.com/blog/best-practices-web-application-development> (дата звернення 23.04.2025).

18. Jielin Y. The Distribution Function of Random Variable and Extension of Mathematical Expectation. URL: <https://iopscience.iop.org/article/10.1088/1742-6596/3004/1/> (дата звернення 23.04.2025).

19. Innovate and transform with a modern application platform. URL: <https://surl.li/khqivi> (дата звернення 23.04.2025).

20. Vako Shvili. What Is Web Design and How Does It Work? URL: <https://surl.li/augojs> (дата звернення 23.04.2025).

21. Beginners Guide to Web Application Development (2025). URL: <https://budibase.com/blog/web-application-development/> <https://surl.li/augojs> (дата звернення 27.04.2025).

					КВРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		59

22. Client-Server Model. URL: <https://www.geeksforgeeks.org/client-server-model/> (дата звернення 29.04.2025).

23. Як будувати UML діаграми. URL: <https://dou.ua/forums/topic/40575/> (дата звернення 29.04.2025).

24. Microservices architecture design. URL: <https://learn.microsoft.com/en-us/azure/architecture/microservices/> (дата звернення 01.05.25).

25. Моделі і методи проєктування інформаційних систем. URL: [https://elearning.sumdu.edu.ua/free\\_content/lectured:de1c9452f2a161439391120eef364dd8ce4d8e5e/20160217112601/183252/index.html](https://elearning.sumdu.edu.ua/free_content/lectured:de1c9452f2a161439391120eef364dd8ce4d8e5e/20160217112601/183252/index.html) (дата звернення 01.05.25).

26. Тестування програмного забезпечення: типи, види та застосування. URL: <https://foxminded.ua/testuvannia-prohramnoho-zabezpechennia/> (дата звернення 15.05.25).

27. Архітектура програмного забезпечення: все що треба знати. URL: <https://wezom.com.ua/ua/blog/arhitektura-programmnogo-obespecheniya> (дата звернення 26.05.2025).

28. Огляд MVC-контролерів ASP.NET. Режим доступу до ресурсу: <https://learn.microsoft.com/en-us/aspnet/mvc/overview/older-versions-1/controllers-and-routing/aspnet-mvc-controllers-overview-cs> (дата звернення 27.05.25).

29. McGrath M. HTML, CSS and JavaScript in easy steps. – Southam. UK: In Easy Steps Limited. 2020. P. 471.

30. Zammetti F. Modern Full-Stack Development: Using TypeScript, React, Node.js, Webpack, and Docker. Frank Zammetti : Apress, 2020. P. 395.

31. Microsoft ASP.NET Identity Documentation. URL: <https://learn.microsoft.com/en-us/aspnet/identity/overview/getting-started/introduction-to-aspnet-identity> (дата звернення 27.05.2025).

32. Цибульник С. О., Барандич К. С. Технології розроблення програмного забезпечення. Частина 1. Життєвий цикл програмного забезпечення. Київ, 2022. 270 с.

33. Tielens M. T. React in Action. Manning Publications. 2021. P. 368.

					КвРІПЗ. 2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		60

34. СОУ 207.01:2017. Текстові документи. Загальні вимоги / Бойко Ю. М., Красильнікова Г. В., Першина Л. І., Косянчук Т. Ф. [2-ге вид., випр.]. Хмельницький : ХНУ, 2018. 45 с.

35. Johnston J. How to Build a Web App: Beginner's Guide. URL: <https://budibase.com/blog/how-to-make-a-web-app/> (дата звернення 27.05.25).

36. Web Application Development: Process, Tools, & Examples. URL: <https://www.browserstack.com/guide/web-application-development-guide> (дата звернення 27.05.25).

37. Unified Modeling Language (UML) Diagrams. URL: <https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/> (дата звернення 28.05.25).

38. Anatomy of the Client/Server Model. URL: [https://docs.oracle.com/cd/E13203\\_01/tuxedo/tux71/html/intbas3.htm](https://docs.oracle.com/cd/E13203_01/tuxedo/tux71/html/intbas3.htm) (дата звернення 28.05.25).

39. What is The Client-Server Architecture? Everything you Need to Know. URL: <https://cybertalents.com/blog/client-server-architecture> (дата звернення 28.05.25).

40. ДСТУ 8302:2015. Бібліографічне посилання. Загальні положення та правила складання. [Уведено вперше; чинний від 2016–07–01]. Вид. офіц. Київ : ДП «УкрНДНЦ», 2016. 17 с.

					КвРІПЗ.2101088.01.15.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		61

**Додаток А**  
(обов'язковий)

**ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ**

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра інженерії програмного забезпечення

**КВАЛІФІКАЦІЙНА РОБОТА**  
НА ТЕМУ:

**«ВЕБЗАСТОСУНОК ДЛЯ РЕАЛІЗАЦІЇ  
МОЛОЧНОЇ ПРОДУКЦІЇ»**

Підготував:  
Студент групи ІПЗ-21-1  
Садовський Артем Олександрович

Керівник кваліфікаційної  
роботи  
асистент  
Дьоміна Анастасія Іванівна

Рисунок А.1 – Титульний слайд, представлення

2-й слайд

## Актуальність теми

Молочна продукція – це специфічний вид продуктів харчування, що є одними з популярних серед населення не тільки України, але й світу. Відомо, що молоко виконує функцію захисного чинника для організму завдяки вмісту в ньому поживних речовин, головним чином, білка, лактози, мінеральних солей, вітамінів тощо.

Хоча електронна комерція може бути доступною не всім переробникам молочної продукції, вона безумовно може бути варіантом для отримання додаткового доходу.

Як показує статистика замовлення молочної продукції, зокрема твердих сирів, йогуртів, кисломолочного сиру в мережі Інтернет через різноманітні вебзастосунки є актуальним та широко використовується.

Рисунок А.2 – Актуальність теми

## 3-й слайд

## Мета та Завдання

**Метою** кваліфікаційної роботи є розробка веб застосунку для продажу молочної продукції, що має можливість реалізувати функціонал для різноманітних категорій користувачів.

**Завдання для роботи:**

- проаналізувати ринок реалізації молочної продукції в мережі Інтернет;
- провести аналітичний огляд існуючих програмно-технічних рішень, а саме веб ресурсів з продажу молокопродуктів із виділенням негативних та позитивних практик реалізації;
- визначити функціональні та нефункціональні вимоги програмного застосунку, яке буде розроблено;
- розробити технічне завдання;
- здійснити вибір типу архітектури та проектування бази даних;
- здійснити реалізацію програмного застосунку для продажу молочної продукції;
- протестувати веб застосунок для реалізації молочної продукції.

Рисунок А.3 – Мета та Завдання

## 4-й слайд

### Змістовий аналіз предметної області, її структурних та функціональних особливостей

Предметна область даної роботи – це ринок реалізації молочної продукції в мережі Інтернет за допомогою онлайн-комерції: вебзастосунків, вебресурсів, маркетплейсів тощо.

Даний застосунок має на меті надати можливість реалізувати свою продукцію не тільки ритейлерам (перепродаж), але й безпосередньо виробникам.

Рисунок А.4 – Змістовий аналіз предметної області, її структурних та функціональних особливостей

## 5-й слайд

## Аналіз наявного програмного-технічного забезпечення

Функціонал	Галичина	Сладосвіт
Простий та зрозумілий дизайн	+	+
Фільтр продуктів	-	+/-
Зображення продукції	+/-	+
Інтуїтивна навігація	+	-
Залежність від плагінів	-	-

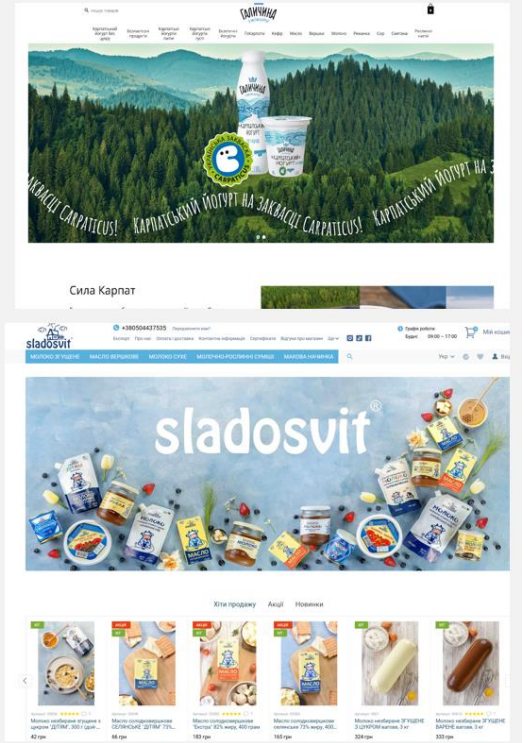


Рисунок А.5 – Аналіз наявного програмного-технічного забезпечення

6-й слайд

## Визначення функціональних та нефункціональних вимог до ПЗ

Функціональні вимоги	Нефункціональні вимоги
<u>Наявність системи реєстрації та авторизації користувачів;</u>	<u>У випадку помилки в оформленні замовлення користувач має отримати зрозуміле повідомлення і можливість повторити дію.</u>
<u>Наявність каталогу товарів та можливості пошуку у ньому;</u>	<u>Система повинна автоматично відновлюватися після перезапуску без втрати даних користувача.</u>
<u>Можливість перегляду деталей кожного товару;</u>	<u>Планове обслуговування повинно проводитися в нічний час і з попередженням користувачів.</u>
<u>Кошик для покупок;</u>	
<u>Оформлення замовлення та вибір способу доставки;</u>	
<u>Можливість залишити відгуків та зворотній зв'язок;</u>	
<u>Наявність ролі адміністратора, тобто можна здійснювати коригування інформації щодо того чи іншого товару.</u>	

Рисунок А.6 – Визначення функціональних та нефункціональних вимог до ПЗ

7-й слайд

## Вибір типу архітектури та шаблонів проектування

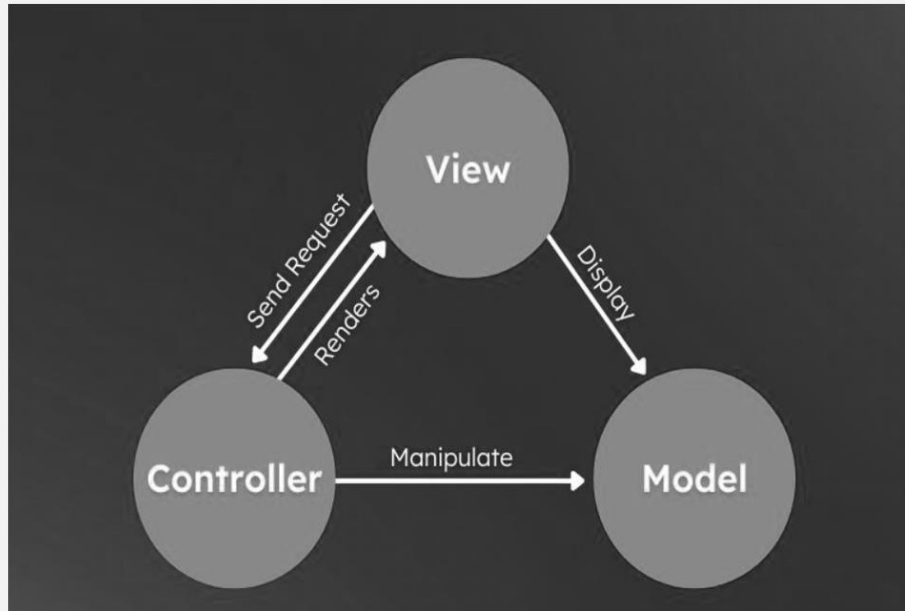


Рисунок А.7 – Вибір типу архітектури та шаблонів проектування

8-й слайд

# Опис декомпозиції

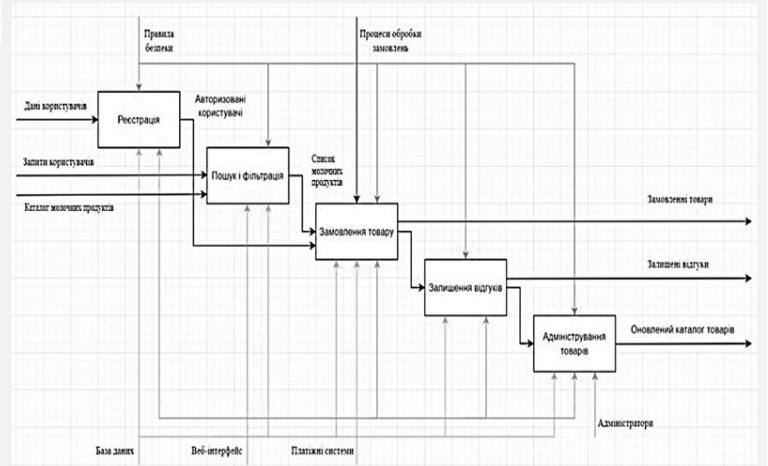


Рисунок А.8 – Опис декомпозиції

## 9-й слайд

## Проектування модулів і даних

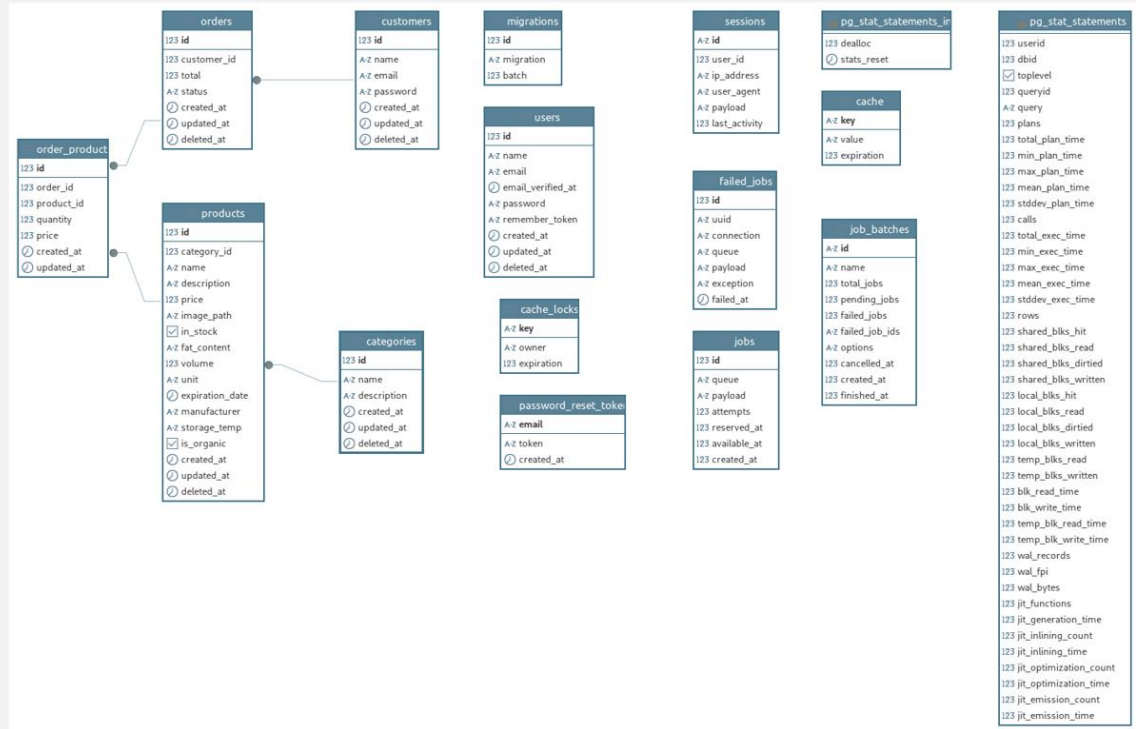


Рисунок А.9 – Проектування модулів і даних

## 10-й слайд

## Аналіз та вибір технологій

**Бекенд:**

- PHP
- [Laravel](#)
- [Composer](#)

**База даних:**

- [PostgreSQL](#)

**Фронтенд:**

- HTML5
- CSS3
- JavaScript ([Vanilla JS](#))
- [Livewire](#): для [інтерактивних компонентів](#)

**Середовище розробки:**

- [Docker](#): для [локальної розробки](#) і деплою
- [Nginx](#): веб сервер
- [Render.com](#): хостинг платформа для [деплойменту](#)

Рисунок А.10 – Аналіз та вибір технологій

11-й слайд

## Програмна реалізація

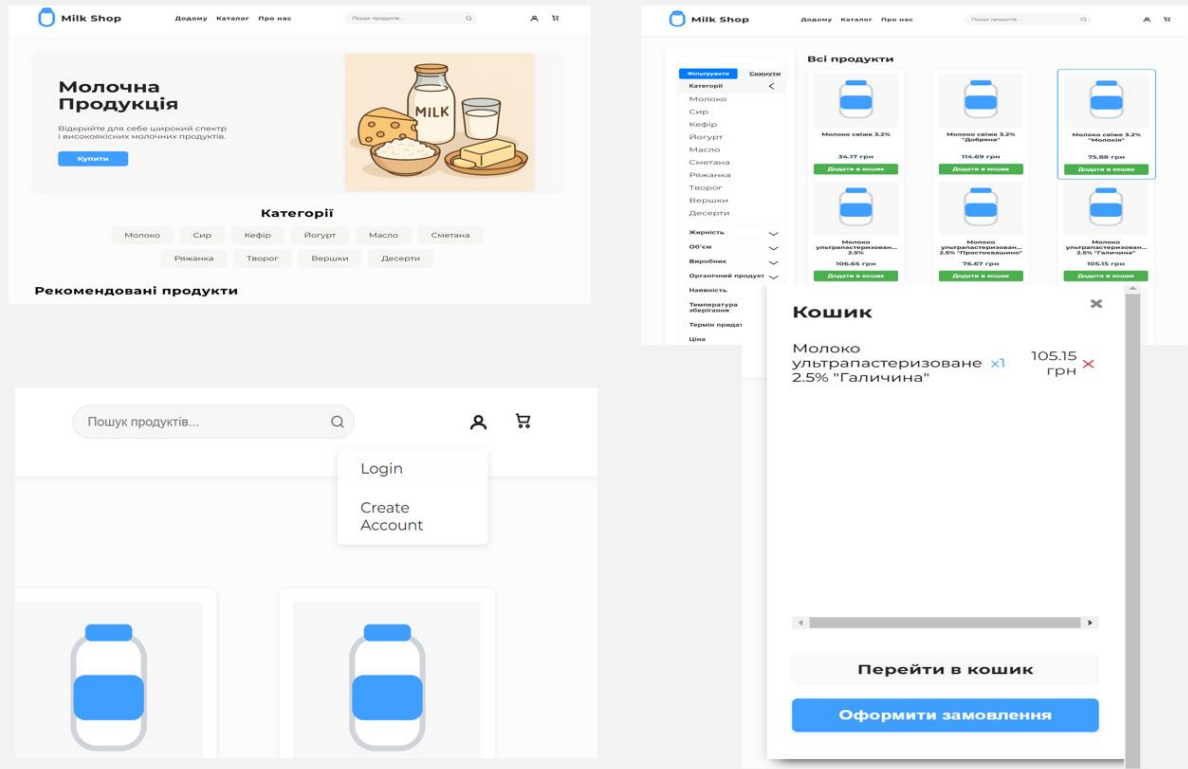


Рисунок А.11 – Програмна реалізація

## 12-й слайд

**Вимоги до технічного та програмного забезпечення**

Операційна система	Windows, macOS, Linux
Браузер	Google Chrome, Mozilla Firefox, Safari, Edge
Дисплей	Мінімум 1280x720 пікселів
Процесор	Двоядерний, 1,8 ГГц
Оперативна пам'ять	2 ГБ
Місце на диску	200 МБ

Рисунок А.12 – Вимоги до технічного та програмного забезпечення

# Тестування ПЗ

The screenshot displays a Jenkins-style log for a deployment process. The log is titled 'milkahop-würt / Logs' and 'Logs'. It shows a series of test results for various features. A red box highlights a failed test in the left column and its summary in the right column.

**Failed Test Details (Left Column):**

```
May 7 10:36:31 PM INFO FAILED Tests\Feature\ProductSearchTest > search products by keyword
Expected: <!DOCTYPE html>\n
<html lang="uk">\n
<head>\n
... (672 more lines)
To contain: Monoko 3.2%

at tests\Feature\ProductSearchTest.php:78
74 | {
75 |     $response = $this->get('/search?query=моніко');
76 |
77 |     $response->assertStatus(200);
+ 78 |     $response->assertSee('Моніко 3.2%');
79 |     $response->assertSee('Моніко зняжкрене 0.5%');
80 |     $response->assertDontSee('Йогурт полуничний');
81 | }
82 |

Tests: 1 failed, 26 passed (75 assertions)
Duration: 9.61s
```

**Test Summary (Right Column):**

```
Tests: 26 passed (69 assertions)
Duration: 9.49s

Some tests failed but continuing deployment as this is a non-blocking check
Test results reported. Deployment continues regardless of test results.
Publishing Livewire assets...
```

Рисунок А.13 – Тестування ПЗ

14-й слайд

## Висновки

Поставлені завдання	Результат виконання
<u>проаналізувати ринок реалізації молочної продукції в мережі Інтернет;</u>	Проведено <u>аналіз сучасних онлайн-платформ з продажу молочної продукції та визначено основні тенденції.</u>
<u>провести аналітичний огляд існуючих програмно-технічних рішень, а саме веб ресурсів з продажу молокопродуктів із виділенням негативних та позитивних практик реалізації;</u>	Оглянуто декілька вебзастосунків, виявлено сильні та слабкі сторони у реалізації функціоналу, дизайну та юзабіліті.
<u>визначити функціональні та нефункціональні вимоги розроблюваного програмного застосунку;</u>	<u>Складено перелік основних функцій і технічних характеристик майбутнього застосунку, включаючи вимоги до безпеки та продуктивності.</u>
<u>розробити технічне завдання;</u>	<u>Підготовлено повне технічне завдання з описом функціоналу, структури бази даних та вимог до системи.</u>
<u>здійснити вибір типу архітектури та проектування бази даних;</u>	<u>Обрано MVC-архітектуру та спроектовано реляційну базу даних для збереження інформації про користувачів і товари.</u>
<u>здійснити реалізацію програмного застосунку для реалізації молочної продукції;</u>	<u>Реалізовано вебзастосунок на основі MVC-архітектури з функціоналом реєстрації, каталогу товарів, кошика та замовлень.</u>
<u>протестувати веб застосунок для реалізації молочної продукції.</u>	<u>Проведено unit-тестування основних модулів програми та усунуено виявлені помилки.</u>

Рисунок А.14 – Висновки

15-й слайд

**Дякую за увагу!**

Рисунок А.15 – Дякую за увагу!

**Додаток Б**  
(обов'язковий)

**ТЕХНІЧНЕ ЗАВДАННЯ**

**Вступ**

Робота виконується в рамках кваліфікаційної роботи на тему «Вебзастосунок для реалізації молочної продукції».

**1 Підстава для розробки**

Підставою для розробки даного програмного забезпечення є «Завдання на кваліфікаційну роботу», що затверджене завідувачем кафедри інженерії програмного забезпечення.

Найменування розробки: «Вебзастосунок для реалізації молочної продукції».

**2 Призначення розробки**

Вебзастосунок для реалізації молочної продукції призначений для забезпечення користувачів можливістю швидкого пошуку, огляду та купівлі молочної продукції.

Користувачами системи є фермери, що бажають продати свій продукт, а також звичайні споживачі, що люблять молокопродукти, тобто вікова категорія може бути від самого малого до найстаршого.

Користувачі можуть легко знайти потрібний товар, використовуючи пряму адресу ресурсу (якщо пошукова оптимізація спрацювала ефективно і сайт вже відомий аудиторії), або за допомогою уточнюючих критеріїв. Для молочної продукції це може бути фірма-виробник, вид продукту та бажаний діапазон ціни.

Крім того, на сайті передбачено форму зворотного зв'язку, можливість додавати товари до кошика, оформлювати замовлення та обирати зручний спосіб оплати та доставки.

Система є доступною для використання на будь-якому персональному комп'ютері та вебпереглядачі без додаткових налаштувань.

### **3 Вимоги до вебзастосунку**

#### **3.1 Вимоги до функціональних характеристик**

Вебзастосунок для продажу молочної продукції повинен передбачати:

- керування асортиментом продукції;
- зручний пошук;
- управління користувачами, а саме реєстрація та авторизація користувачів.
- персоналізація профілю;
- адміністрування.

Вимоги до дизайну інтерфейсу користувача полягають у створенні чистого, сучасного та легкого у використанні середовища. Критично важливим є забезпечення актуальності всієї інформації про продукцію, щоб користувачі завжди отримували свіжі та достовірні дані.

#### **3.2 Вимоги до надійності програмного забезпечення**

Розроблюваний вебзастосунок має мати:

- відновлення після технічних перебоїв;
- захист від несанкціонованого доступу;
- керування правами;
- перевірка введених даних.

У разі збою (наприклад, через втрату живлення) система здатна самостійно відновити свою працездатність. Дані користувачів та системи надійно захищені від стороннього втручання. Для кожного користувача чітко визначені дозволи на доступ до різних частин системи. Система контролює коректність вхідної інформації та запобігає неправомірним діям користувачів.

#### **3.3 Вимоги до складу та параметрів технічних засобів**

Мінімальні вимоги для роботи вебзастосунку мають відповідати вимогам будь-якого переглядача. Далі подано список мінімальних вимог для переглядача Google Chrome:

- безперебійне з'єднання з мережею Інтернет;
- ОС Windows 10, 8, 8.1, 7;

- розрядність - x64 (64-bit);
- процесор: Pentium 4 з SSE2;
- вінчестер: мінімум 350 Mb;
- відеоадаптер: 3D адаптер nVidia, Intel, AMD / ATI;
- оперативна пам'ять: мінімум 512 Mb;
- роздільна здатність екрану: SVGA 800x600;
- периферійні пристрої: клавіатура, миша;
- будь-яка аудіокарта;

### **3.4 Вимоги до інформаційної та програмної сумісності**

Бекенд:

- PHP 8.2.28;
- Laravel 12.12.0;
- Composer 2.5;
- База даних PostgreSQL 14.

Фронтенд:

- HTML5
- CSS3
- JavaScript (Vanilla JS)
- Livewire: для інтерактивних компонентів

Середовище розробки:

- для локальної розробки і деплою використано Docker;
- вебсервер Nginx;
- хостинг платформа для деплойменту Render.com.

### **3.5 Вимоги до транспортування та зберігання**

Програмне забезпечення та вся необхідна документація надаються в електронному вигляді. Умови функціонування програми залежать від умов експлуатації сервера, де вона буде розгорнута.

### **3.6 Спеціальні вимоги**

Застосунок вимагає дружнього інтерфейсу, який буде зрозумілий для користувачів із середньою кваліфікацією у сфері інформаційних технологій. Визначення мови програмування є прерогативою розробника.

#### **4 Загальні вимоги до програмної документації**

В процесі розробки програмного забезпечення мають бути підготовлені:

- ТЗ або специфікація;
- загальний опис програми;
- коментарі та пояснення в тексті програмного забезпечення;
- методика здійснення випробувань;
- відомості про функціонал програмного забезпечення;
- інструкція для користувача;
- інструкція для розробника.

#### **5 Стадії та етапи розробки**

Розробка програмного продукту проходить декілька стадій і етапів:

- визначення вимог;
- первинне проектування (ескізний проект);
- детальне проектування (технічний проект);
- підготовка до кодування (робочий проект);
- створення супровідної документації;
- проведення перевірки функціоналу;
- введення в експлуатацію.

#### **6 Порядок контролю та приймання**

Контроль якості вебзастосунку здійснюється кінцевими користувачами, залученими на етапі тестування. Приймання системи відбувається лише після її повного розгортання на хостингу та налаштування для коректного функціонування.

Крім того, після завершення розробки, обов'язково проводиться тестування на захист від некоректного введення даних.

## ДОДАТОК В (обов'язковий)

### КОД ПРОГРАМИ

```
server {  
    listen 80;  
    index index.php index.html;  
    server_name localhost;  
    root /var/www/src/public;  
  
    location / {  
        try_files $uri $uri/ /index.php?$query_string;  
    }  
  
    location ~ /\.php$ {  
        fastcgi_pass app:9000;  
        fastcgi_index index.php; <?php  
  
namespace App\Filament\Resources\CategoryResource\Pages;  
  
use App\Filament\Resources\CategoryResource;  
use Filament\Actions;  
use Filament\Resources\Pages\CreateRecord;  
  
class CreateCategory extends CreateRecord  
{  
    protected static string $resource = CategoryResource::class;  
}  
  
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;  
        include fastcgi_params;  
        fastcgi_param HTTP_X_FORWARDED_FOR $proxy_add_x_forwarded_for;  
        fastcgi_param HTTP_X_FORWARDED_PROTO $scheme;
```

```
    }

    location ~ /\.ht {
        deny all;
    }
}
<?php

namespace App\Filament\Resources\CategoryResource\Pages;

use App\Filament\Resources\CategoryResource;
use Filament\Actions;
use Filament\Resources\Pages>EditRecord;

class EditCategory extends EditRecord
{
    protected static string $resource = CategoryResource::class;

    protected function getHeaderActions(): array
    {
        return [
            Actions\DeleteAction::make(),
        ];
    }
}
<?php

namespace App\Filament\Resources\CategoryResource\Pages;

use App\Filament\Resources\CategoryResource;
use Filament\Actions;
use Filament\Resources\Pages>ListRecords;

class ListCategories extends ListRecords
{
    protected static string $resource = CategoryResource::class;
```

```

protected function getHeaderActions(): array
{
    return [
        Actions\CreateAction::make(),
    ];
}
}
<?php

namespace App\Filament\Resources\CategoryResource\RelationManagers;

use Filament\Forms;
use Filament\Forms\Form;
use Filament\Resources\RelationManagers\RelationManager;
use Filament\Tables;
use Filament\Tables\Table;
use Illuminate\Database\Eloquent\Builder;
use Illuminate\Database\Eloquent\SoftDeletingScope;

class ProductsRelationManager extends RelationManager
{
    protected static string $relationship = 'products';

    public function form(Form $form): Form
    {
        return $form
            ->schema([
                Forms\Components\TextInput::make('name')
                    ->required()
                    ->maxLength(255),
                Forms\Components\Textarea::make('description')
                    ->columnSpanFull(),
                Forms\Components\TextInput::make('price')
                    ->required()
                    ->numeric()
            ])
    }
}

```

```

        ->prefix('$'),
        Forms\Components\Toggle::make('in_stock')
        ->required(),
    ]);
}

public function table(Table $table): Table
{
    return $table
        ->recordTitleAttribute('name')
        ->columns([
            Tables\Columns\TextColumn::make('name')
                ->searchable(),
            Tables\Columns\TextColumn::make('price')
                ->money('USD')
                ->sortable(),
            Tables\Columns\IconColumn::make('in_stock')
                ->boolean(),
            Tables\Columns\TextColumn::make('created_at')
                ->dateTime()
                ->sortable()
                ->togglable(isToggledHiddenByDefault: true),
        ])
        ->filters([
            Tables\Filters\TrashedFilter::make()
        ])
        ->headerActions([
            Tables\Actions\CreateAction::make(),
        ])
        ->actions([
            Tables\Actions\EditAction::make(),
            Tables\Actions\DeleteAction::make(),
            Tables\Actions\ForceDeleteAction::make(),
            Tables\Actions\RestoreAction::make(),
        ])
        ->bulkActions([

```

```

        Tables\Actions\BulkActionGroup::make([
            Tables\Actions\DeleteBulkAction::make(),
            Tables\Actions\ForceDeleteBulkAction::make(),
            Tables\Actions\RestoreBulkAction::make(),
        ]),
    ]),
    ->modifyQueryUsing(fn (Builder $query) => $query-
->withoutGlobalScopes([
        SoftDeletingScope::class,
    ]));
    }
}
<?php

```

```
namespace App\Filament\Resources\CustomerResource\Pages;
```

```
use App\Filament\Resources\CustomerResource;
```

```
use Filament\Actions;
```

```
use Filament\Resources\Pages\CreateRecord;
```

```
class CreateCustomer extends CreateRecord
```

```
{
```

```
    protected static string $resource = CustomerResource::class;
```

```
}
```

```
<?php
```

```
namespace App\Filament\Resources\CustomerResource\Pages;
```

```
use App\Filament\Resources\CustomerResource;
```

```
use Filament\Actions;
```

```
use Filament\Resources\Pages>ListRecords;
```

```
class ListCustomers extends ListRecords
```

```
{
```

```
    protected static string $resource = CustomerResource::class;
```

```

protected function getHeaderActions(): array
{
    return [
        Actions\CreateAction::make(),
    ];
}
}
<?php

namespace App\Filament\Resources\CustomerResource\RelationManagers;

use Filament\Forms;
use Filament\Forms\Form;
use Filament\Resources\RelationManagers\RelationManager;
use Filament\Tables;
use Filament\Tables\Table;
use Illuminate\Database\Eloquent\Builder;
use Illuminate\Database\Eloquent\SoftDeletingScope;

class OrdersRelationManager extends RelationManager
{
    protected static string $relationship = 'orders';

    public function form(Form $form): Form
    {
        return $form
            ->schema([
                Forms\Components\TextInput::make('customer_id')
                    ->required()
                    ->maxLength(255),
            ]);
    }

    public function table(Table $table): Table
    {
        return $table

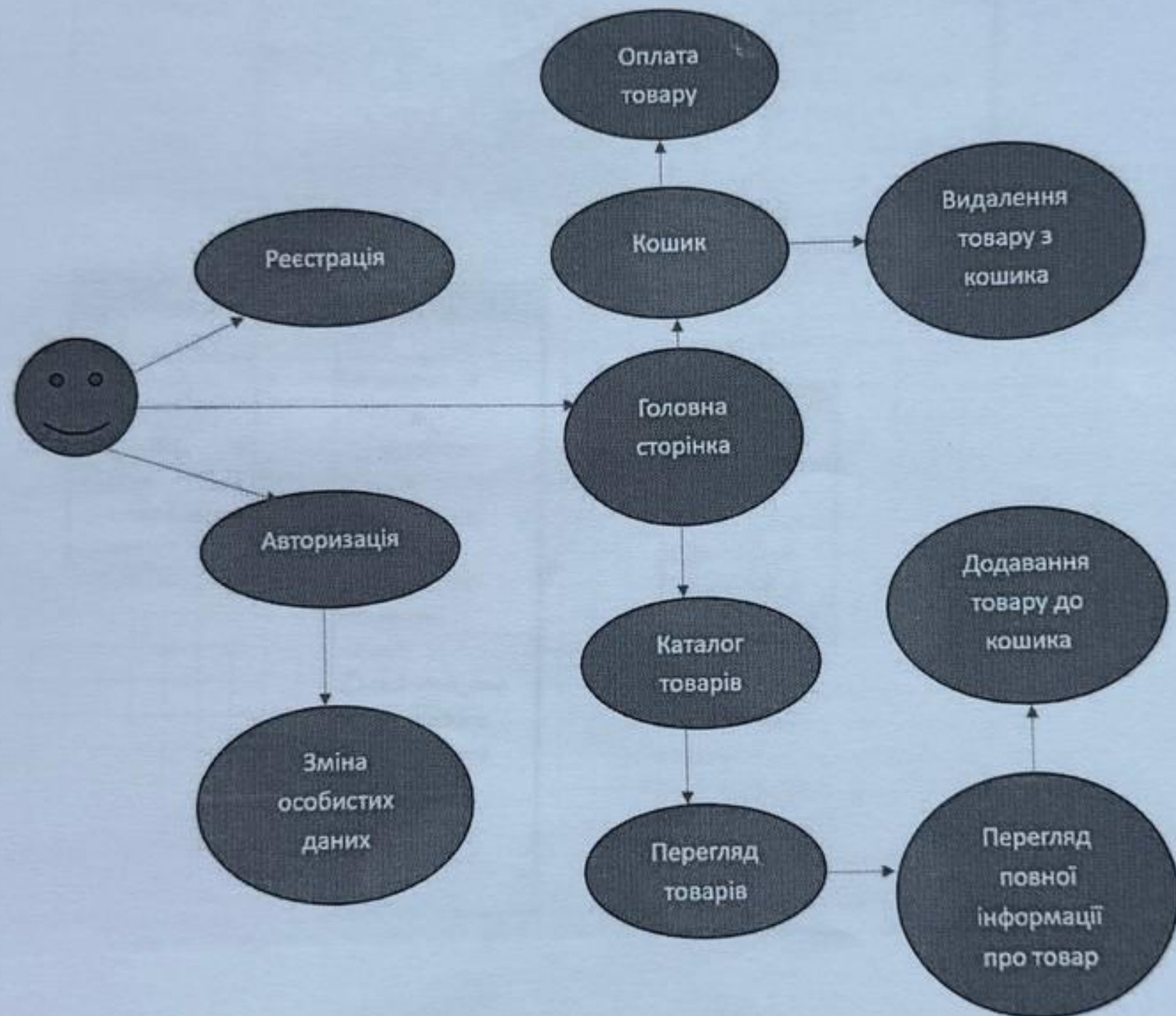
```

```

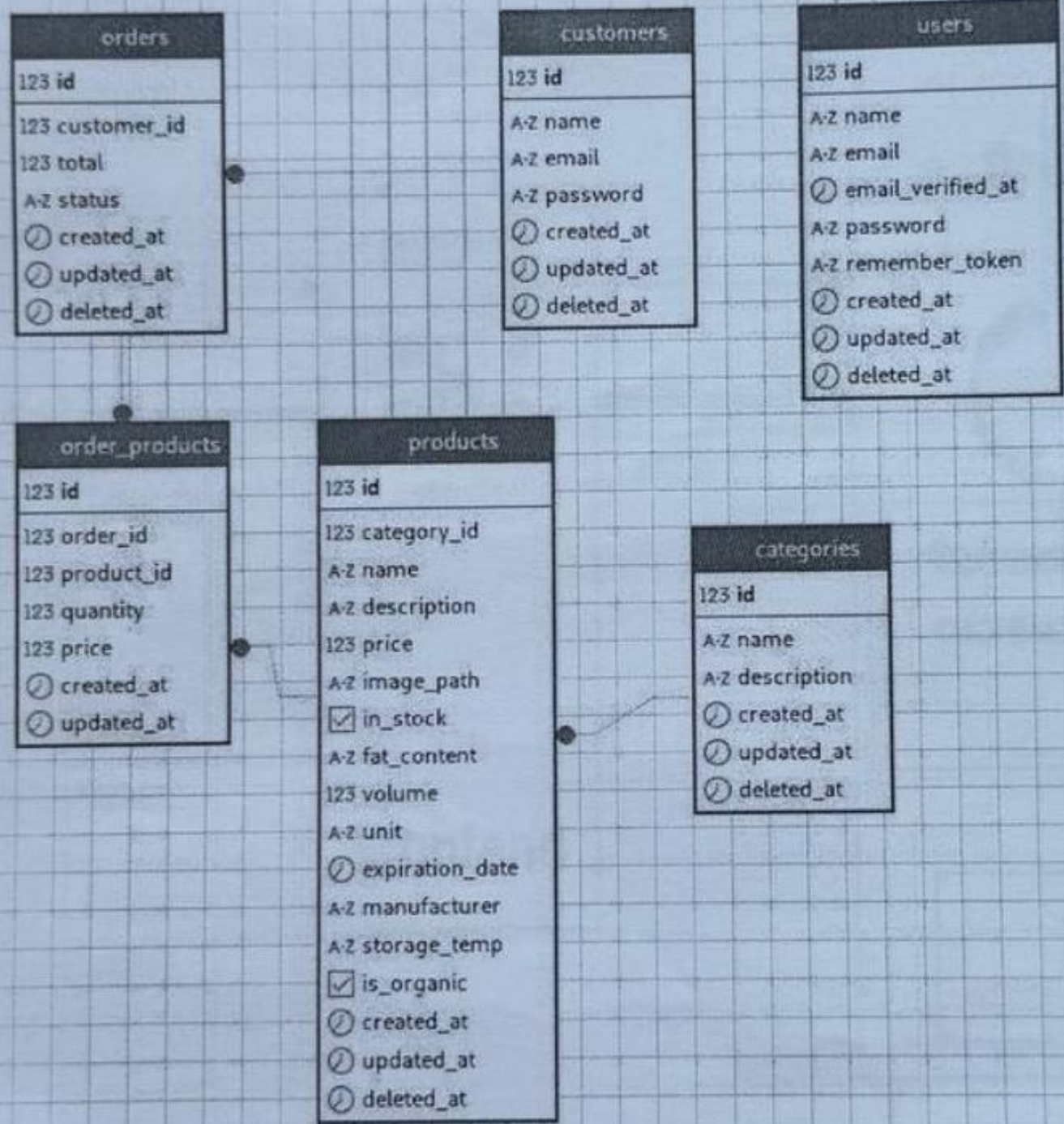
->recordTitleAttribute('customer_id')
->columns([
    Tables\Columns\TextColumn::make('customer_id'),
])
->filters([
    Tables\Filters\TrashedFilter::make()
])
->headerActions([
    Tables\Actions\CreateAction::make(),
])
->actions([
    Tables\Actions\EditAction::make(),
    Tables\Actions\DeleteAction::make(),
    Tables\Actions\ForceDeleteAction::make(),
    Tables\Actions\RestoreAction::make(),
])
->bulkActions([
    Tables\Actions\BulkActionGroup::make([
        Tables\Actions\DeleteBulkAction::make(),
        Tables\Actions\ForceDeleteBulkAction::make(),
        Tables\Actions\RestoreBulkAction::make(),
    ]),
])
->modifyQueryUsing(fn (Builder $query) => $query-
>withoutGlobalScopes([
    SoftDeletingScope::class,
]));
}
}

```

## **ГРАФІЧНА ЧАСТИНА**

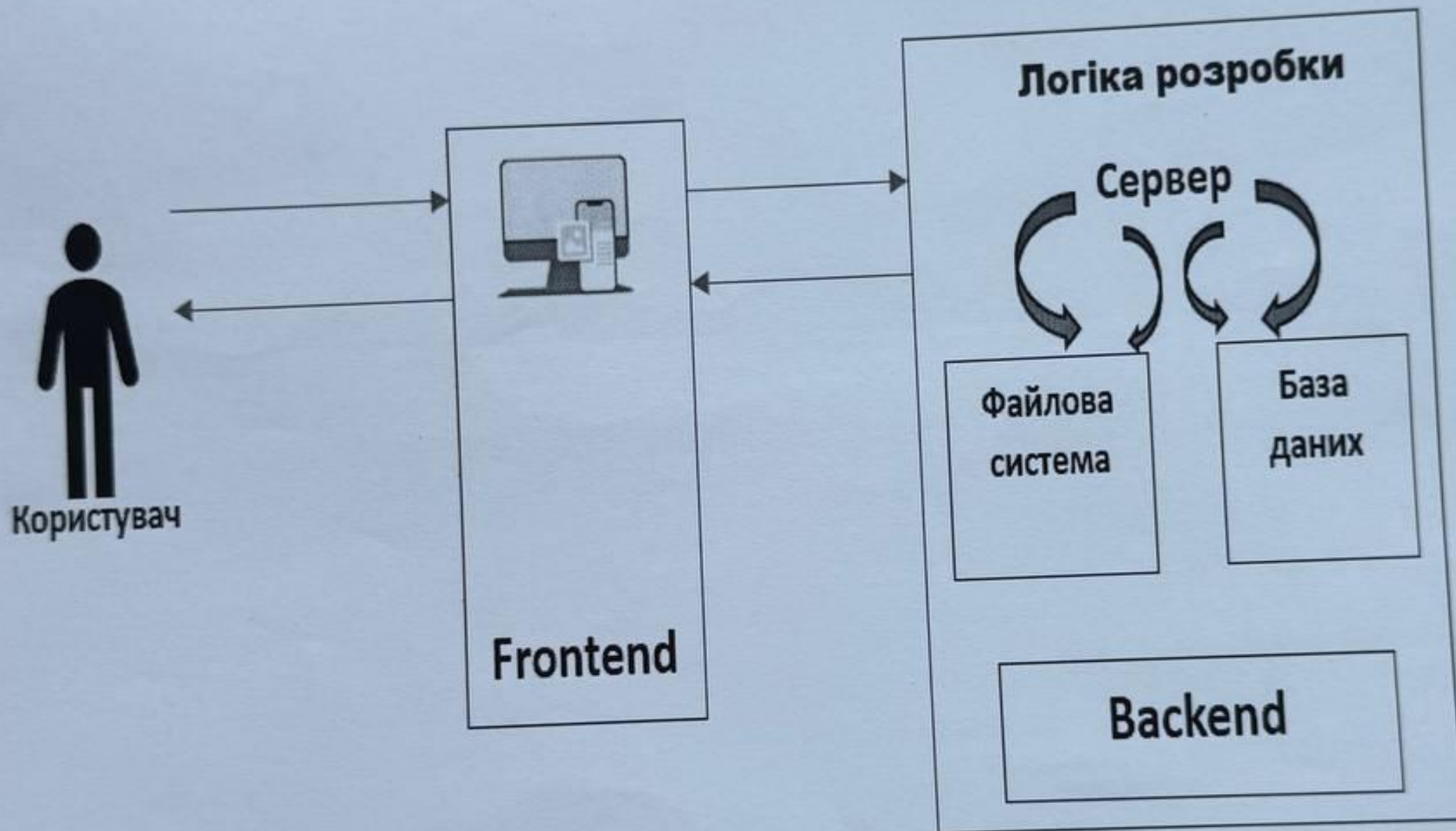


				КвРІПЗ. 2101088.01.15.E8		
				Вебзастосунок для реалізації молочної продукції		
				Літера		
				Маса		
				Масштаб		
Зм.	Арк.	№ докум.	Підпис	Діаграма варіантів використання		
Розробив	Садовський					
Керівник	Дьоміна А.І.					
Консульт.						
Н. Конт.	Яшина О.М.			Аркуш 1   Аркушів 3		
Зав. каф.	Бедратюк Л. П.					
				ХНУ, ІПЗ-22-1		



					КвРІПЗ. 2101088.01.18.E8			
					Вебзастосунок для реалізації молочної продукції			
					Схема бази даних			
					Літера		Маса	Масштаб
					Аркуш 2		Аркушів 3	
					ХНУ, ІПЗ-22-1			

Зм. Арк.	№ докум.	Підпис	Дата
Розробив	Садовський	<i>[Signature]</i>	18.10.2018
Керівник	Дьоміна А.І.	<i>[Signature]</i>	18.10.2018
Консульт.			
Н. Контр.	Яшина О.М.	<i>[Signature]</i>	18.10.2018
Зав. каф.	Бедратюк Л. П.	<i>[Signature]</i>	18.10.2018



Зм.	Арк.	№ докум.	Підпис	Дата
Розробив		Садовський	<i>[Signature]</i>	2015
Керівник		Дьоміна А.І.	<i>[Signature]</i>	2015
Консульт.				
Н. Контр.		Яшина О.М.	<i>[Signature]</i>	2015
Зав. каф.		Белратюх Л. П.	<i>[Signature]</i>	2015

КвРІПЗ. 2101088.01.15.E8			
Вебзастосунок для реалізації молочної продукції	Літера	Маса	Масштаб
Архітектура	Аркуш 3	Аркушів 3	
ХНУ, ІПЗ-21-1			

## **СУПРОВІДНІ ДОКУМЕНТИ**

Завідувачу кафедри інженерії програмного  
забезпечення проф. Леоніду БЕДРАТЮКУ  
здобувача вищої освіти  
Садовський Артем Олександрович  
факультет ІТ, ІVкурс, група ІПЗ-21-1

### ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності в Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а) та надаю свою згоду на обробку й збереження університетом моєї роботи в інституційному репозитарії Хмельницького національного університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-обчислювального комплексу StrikePlagiarism та/або програмно-технічного засобу AntiPlagiarism і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення текстових збігів у роботах.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

02.01.2025  
дата

С.К.  
підпис

## Anti-Plagiarism (UA) v-15.281 Educational

**The maximum coincidence with one document 3.0%**

**Dictionaries check: en\_US, ru\_RU, ua\_UA. Errors in the documents: 13%**

ID: 244352 Title: БКР_Вебзастосунок для реалізації молочної продукції Added in a DB: 2025-06-09 Authors: Садовський Артем Heads: ДЬОМІНА Анастасія Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	48191	758	3620 (8%)	57 (8%)

### Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

## Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

**Автор:** Садовський Артем

**Співавтор:**

**Назва:** БКР\_Вебзастосунок для реалізації  молочної  продукції

**Науковий керівник:**

**Підрозділ:** Кафедра інженерії програмного забезпечення

**Коефіцієнт подібності 1:**6.2%

**Коефіцієнт подібності 2:**1.2%

**Мікропробіли:** 0

**Заміна букв:** 3

**Інтервали:** 0

**Білі знаки:** 0

**Дата створення звіту:** 2025-06-09 16:59:46.0

**Після аналізу Звіту подібності констатую наступне:**

**Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

**Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

**Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.**

**Обґрунтування:**

Дата 09.06.2025

експерт



## ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ  
освітнього ступеня «Бакалавр»Дипломник Садовський Артем ОлександровичТема Вебзастосунок для реалізації молочної продукціїСпеціальність 121 – Інженерія програмного забезпечення**Обсяг кваліфікаційної роботи:**Кількість листів креслень 3; кількість сторінок записки 84

1. Короткий зміст пояснювальної записки та прийнятих рішень У кваліфікаційній роботі було досліджено та проаналізовано сферу реалізації молочної продукції, визначено всі функціональні та нефункціональні вимоги до вебзастосунку. Проведено аналіз існуючих рішень на ринку, виявлено їх сильні та слабкі сторони, що підтвердило доцільність створення нового вебзастосунку на ринку сучасних технологій та інструментів, які були використані для реалізації проекту. У результаті розроблено вебзастосунок для продажу молочної продукції. Проведена тестова підтвердило коректність роботи та його готовність до використання в реальних умовах.

2. Висновок про відповідність роботи поставленому завданню Кваліфікаційна робота виконана відповідно до поставленого завдання та з дотриманням всіх вимог.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки та передових методів роботи У вступі доведено актуальність теми, визначено мету та завдання дипломного проектування. У першому розділі проведено аналіз предметної області, розглянуто існуючі рішення та визначені функціональні і нефункціональні вимоги до розроблюваного програмного забезпечення. У другому розділі проведено аналіз сучасних архітектур, розглянуто їх переваги і недоліки та визначено, що система буде відповідати монолітній архітектурі та моделі MVC. У третьому розділі підготовлено всі залежності для написання коду та виконано практичну розробку програмних модулів і описано їх особливості, в результаті чого створено програмний продукт. Також у цьому розділі виконано модульне тестування системи та проведено його у відповідності до функціональних вимог, в результаті було підтверджено коректну роботу програми.

4. Позитивні сторони роботи Робота демонструє ґрунтовний аналіз предметної області, що включає розуміння ролі вебзастосунків у сфері продажу молочної продукції, а також аналіз існуючих рішень із виділенням позитивних практик.

5. Негативні сторони роботи У роботі бракує прикладів коду, опису тригерів. Тестування описано загально, без конкретних тест-кейсів і результатів. Висновки розділів є поверхневими.

6. Оцінка графічного оформлення та пояснювальної записки Графічне оформлення виконано відповідно до теми кваліфікаційної роботи та подано у вигляді діаграм і рисунків. Пояснювальна записка оформлена згідно вимог чинних стандартів.

7. Відгук про кваліфікаційну роботу в цілому Кваліфікаційна робота заслуговує позитивної оцінки. Матеріал пояснювальної записки структурований, послідовний, чіткий та простий, що дозволяє чітко зрозуміти викладений матеріал у рамках тематики проектування. Графічний матеріал дає можливість наочно побачити деталі проектування системи.


8. Інші зауваження \_\_\_\_\_

9. Оцінка кваліфікаційної роботи Кваліфікаційна робота виконана у повному обсязі, відповідає поставленій задачі та заслуговує на оцінку «добре».

РЕЦЕНЗЕНТ

к. т. н., доцент кафе. К/ІС, Камуєсен Марія Вікторівна

“02” червня 2025 р.

  
(підпис)

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ  
КАФЕДРИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатами звіту/звітів перевірки роботи, продуктованими програмно-технічним засобом (ами), на наявність текстових збігів.

Назва кваліфікаційної роботи: «Вебзастосунок для реалізації молочної продукції»

Автор: Садовський Артем Олександрович

Освітня програма: Освітньо-професійна програма «Інженерія програмного забезпечення»

Спеціальність: 121 – Інженерія програмного забезпечення

Науковий керівник: Дьоміна Анастасія Іванівна, асистент

Після аналізу звіту/звітів зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається до захисту.	<b>відповідає</b>
2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована.	
3	Виявлені запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Виявлені запозичення частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнуті. Робота може бути допущена до захисту після того, як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укріття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) у тексті кваліфікаційної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень у стандартних бланках (титулка, завдання, анотація, відомість документів), у структурі змісту, назвах розділів/підрозділів тощо, у назвах публікацій та у переліку джерел посилання;

2) в якості запозичень системою було зафіксовано деякі послідовності вихідного коду і посилання на бібліотеки, які є стандартними мовними конструкціями програмування та не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;

3) усі запозичення є фрагментарними або мають належним чином оформлені посилання;

З урахуванням наведеного, визначений системою виявлення збігів ідентичності/схожості, складає 6,2%, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

Дата 10.06.25

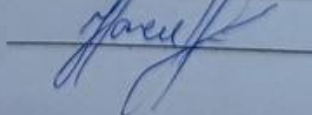
Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи







Леонід БЕДРАТЮК

Леонід БЕДРАТЮК

Анастасія Дьоміна