

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Відокремлений структурний підрозділ
«Тернопільський фаховий коледж
Тернопільського національного технічного університету імені Івана Пулюя»
Відділення телекомунікацій та електронних систем
Циклова комісія комп'ютерних наук**

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

фахового молодшого бакалавра

**на тему: Розробка інтернет-магазину крафтових виробів
«Лісова Майстерня»**

Виконав: студент IV курсу, групи КН-423
спеціальності: 122 Комп'ютерні науки

Назар ГАРДАМАЛА

Керівник _____ Віктор Олександр ТАРАСЮК

Рецензент _____
(ім'я та прізвище)

ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ТЕРНОПІЛЬСЬКИЙ ФАХОВИЙ КОЛЕДЖ
ТЕРНОПІЛЬСЬКОГО НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ ІМЕНІ ІВАНА ПУЛЮЯ»

Відділення телекомунікацій та електронних систем
Циклова комісія комп'ютерних наук
Освітньо-професійний ступінь «фаховий молодший бакалавр»
Спеціальність 122 «Комп'ютерні науки»
Галузь знань 12 «Інформаційні технології»

ЗАТВЕРДЖУЮ

Голова циклової комісії
комп'ютерних наук
_____ Галина МАРЦІЯШ
« _____ » _____ 2026 року

**З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Гардамалі Назару Юрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка інтернет-магазину крафтових виробів «Лісова Майстерня»

керівник роботи Тарасюк Віктор Олександр Мар'янович

(прізвище, ім'я, по батькові)

затверджені наказом вищого навчального закладу № 4/9-132 від 27.02.2026 р.

2. Строк подання студентом роботи: 19.06.2026 р.

3. Вихідні дані до роботи: технічне завдання на розробку програмного забезпечення, мови програмування: Java, JavaScript, JSX, SQL, HTML, SCSS; стандарти IEEE 830-1998, ISO/IEC/IEEE 29148:2018, ISO/IEC/IEEE 29119, ГОСТ 34.602-89, ДСТУ 8302:2015.

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

- 1 Загальний розділ
 - 1.1 Аналітичний огляд існуючих рішень
 - 1.2 Технічне завдання
 - 1.2.1 Найменування та область застосування
 - 1.2.2 Призначення розробки
 - 1.2.3 Вимоги до функціоналу web-сайту
 - 1.2.4 Вимоги до програмної документації
 - 1.2.5 Техніко-економічні показники
 - 1.2.6 Стадії та етапи розробки
 - 1.2.7 Порядок тестування та прийому
- 2 Розробка технічного та робочого проєкт

2.1 Розробка структури сайту і web-сторінок

2.2 Створення та верстка сторінок сайту

2.3 Створення та верстка сторінок сайту

2.4 Програмування сайту

2.4.1 Написання клієнтської частини

2.4.2 Написання admin- частини

2.5 Тестування web- сайту

3 Спеціальний розділ

3.1 Інструкція з розміщення сайту в Інтернеті

3.2 Інструкція з обслуговування та наповнення сайту

3.3 Інструкція з популяризації та підтримки сайту

4 Економічний розділ

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення НДР

4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

4.3 Розрахунок витрат на електроенергію

4.4 Розрахунок суми амортизаційних відрахувань для інтернет-магазину крафтових виробів «Лісова Майстерня»

4.5 Обчислення накладних витрат

4.6 Складання кошторису витрат та визначення собівартості інтернет-магазину крафтових виробів «Лісова Майстерня»

4.7 Розрахунок ціни розробки інтернет-магазину крафтових виробів «Лісова Майстерня»

4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень

5 Охорона праці, техніка безпеки та екологічні вимоги

5.1 Основні заходи по запобіганню травматизму та професійних захворювань.

5.2 Визначення параметрів мікроклімату

6 Висновки (навести результати роботи по кожному розділу зокрема і загальний висновок по кваліфікаційній роботі)

Додаткові вказівки:

Виконання роботи із розробкою програмного продукту інтернет-магазину крафтових виробів «Лісова Майстерня»

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, наприклад):

1. Структурна схема
2. Блок – схема процедури
3. ER діаграма баз даних
4. Таблиця техніко-економічних показників

6. Консультанти розділів роботи

Розділ	Ім'я та прізвище, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний розділ	Любов КАЛУШКА		
Охорона праці, техніка безпеки та екологічні вимоги	Генадій ГОЧЯЧЕК		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту	Строк виконання етапів роботи	Примітка
1	Отримання і аналіз технічного завдання	20.03.2026	
2	Збір і узагальнення інформації по роботі	01.05.2026	
3	Написання першого розділу	15.05.2026	
4	Розробка технічного та робочого проєкту	29.05.2026	
5	Написання спеціального розділу	05.06.2026	
6	Розрахунок економічної частини	08.06.2026	
7	Написання розділу охорони праці	09.06.2026	
8	Виконання графічної частини	10.06.2026	
9	Оформлення роботи	11.06.2026	
10	Погодження нормоконтролю	12.06.2026	
11	Попередній захист роботи	.06.2026	
12	Захист роботи	.06.2026	

7. Дата видачі завдання: 05 березня 2026.

Студент
(підпис)

(ім'я та прізвище)

Назар ГАРДАМАЛА

Керівник роботи
(підпис)

(ім'я та прізвище)

Віктор ТАРАСЮК

ЗМІСТ

АНОТАЦІЯ	6
ВСТУП.....	8
1 ЗАГАЛЬНИЙ РОЗДІЛ	10
1.1 Аналітичний огляд існуючих рішень	10
1.2 Технічне завдання.....	14
1.2.1 Найменування та область застосування.....	15
1.2.2 Призначення розробки.....	16
1.2.3 Вимоги до функціоналу web-сайту	17
1.2.4 Вимоги до програмної документації	20
1.2.5 Техніко-економічні показники.....	21
1.2.6 Стадії та етапи розробки.....	22
1.2.7 Порядок тестування та прийому	25
2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ	26
2.1 Розробка структури сайту і web-сторінок.....	26
2.2 Створення та верстка сторінок сайту	32
2.3 Розробка структури бази даних сайту	38
2.4 Програмування сайту	43
2.4.1 Написання клієнтської частини	46
2.4.2 Написання admin-частини	49
2.5 Тестування web-сайту	53
3 СПЕЦІАЛЬНИЙ РОЗДІЛ	59
3.1 Інструкція з розміщення сайту в Інтернеті.....	59
3.2 Інструкція з обслуговування та наповнення сайту	64
3.3 Інструкція з популяризації та підтримки сайту	67
4 ЕКОНОМІЧНИЙ РОЗДІЛ	70

					2026.КВР.122.423.03.00.00 ПЗ			
Зм.	Арк.	№ докум.	Підпис	Дата	Розробка інтернет-магазину крафтових виробів «Лісова Майстерня» Пояснювальна записка	Лім.	Арк.	Аркушів
Розроб.		Гардамала Н. Ю.					5	105
Перевір.		Тарасюк В. О. М.						
Реценз.								
Н. Контр.								
Затверд.						ВСП ТФК ТНТУ КН-423п м. Тернопіль		

4.1	Визначення стадій технологічного процесу та загальної тривалості проведення НДР	71
4.2	Визначення витрат на оплату праці та відрахувань на соціальні заходи	73
4.3	Розрахунок витрат на електроенергію	75
4.4	Розрахунок суми амортизаційних відрахувань для інтернет-магазину крафтових виробів «Лісова Майстерня»	76
4.5	Обчислення накладних витрат	77
4.6	Складання кошторису витрат та визначення собівартості інтернет-магазину крафтових виробів «Лісова Майстерня»	78
4.7	Розрахунок ціни розробки інтернет-магазину крафтових виробів «Лісова Майстерня»	79
4.8	Визначення економічної ефективності і терміну окупності капітальних вкладень	80
5	ОХОРОНА ПРАЦІ	81
5.1	Основні заходи по запобіганню травматизму та професійних захворювань	81
5.2	Визначення параметрів мікроклімату	86
	ВИСНОВКИ	90
	ПЕРЕЛІК ПОСИЛАНЬ	92
	ДОДАТКИ	94
	Додаток А. App.jsx	94
	Додаток Б. ProductList.jsx	96
	Додаток В. config.js	98
	Додаток Г. UserController.java	99
	Додаток Д. ProductController.java	101
	Додаток Е. CustomOrderController.java	102
	Додаток Ж. AuthFilter.java	104

АНОТАЦІЯ

Тема кваліфікаційної роботи: Розробка інтернет-магазину крафтових виробів «Лісова Майстерня».

Метою роботи є розробка web-сайту інтернет-магазину крафтових виробів, що забезпечує перегляд каталогу, оформлення замовлень, подання індивідуальних заявок, авторизацію користувачів та адміністрування даних.

Пояснювальна записка складається з шести розділів. У роботі виконано аналіз існуючих рішень, сформовано технічне завдання, розроблено структуру сайту, базу даних, клієнтську, серверну й адміністративну частини та проведено тестування. Клієнтська частина реалізована з використанням React та Vite, серверна частина - на основі Spring Boot, база даних - PostgreSQL.

Обсяг пояснювальної записки - _____ сторінок.

Графічна частина містить структурну схему програми, блок-схему процедури та ER-діаграму бази даних.

Qualification work topic: Development of an online store for craft products “Lisova Maisternia”.

The purpose of the work is to develop a website for an online store of craft products that provides catalog browsing, order placement, submission of individual requests, user authorization, and data administration.

The explanatory note consists of six sections. The work includes analysis of existing solutions, technical specification, website structure, database, client, server and administrative parts, and testing. The client part is implemented using React and Vite, the server part is based on Spring Boot, and PostgreSQL is used as the database.

The explanatory note contains _____ pages.

The graphical part includes the program structure diagram, procedure flowchart, and database ER diagram.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

У сучасних умовах розвитку інформаційних технологій web-сайти стали важливим інструментом представлення, просування та реалізації товарів і послуг. Особливо актуальним це є для малого бізнесу, творчих майстерень і виробників крафтової продукції, які потребують зручного способу взаємодії з клієнтами, демонстрації асортименту, приймання замовлень та організації роботи. Власний інтернет-магазин дає змогу підприємству не залежати повністю від соціальних мереж чи сторонніх торговельних платформ, а формувати власний цифровий простір для продажу продукції.

Темою кваліфікаційної роботи є розробка програмного продукту інтернет-магазину крафтових виробів «Лісова Майстерня». Проєкт орієнтований на майстерню, що займається виготовленням дерев'яних виробів ручної роботи, продажем готових товарів і прийманням індивідуальних замовлень. Такий формат діяльності потребує не лише каталогу продукції, а й можливості оформлення заявок за індивідуальними параметрами, перегляду історії замовлень, керування їх статусами та адміністрування асортименту.

Актуальність роботи полягає в тому, що сучасний покупець очікує швидкого доступу до інформації про товар, зручного оформлення замовлення та можливості відстеження його стану. Для крафтових виробів це особливо важливо, оскільки клієнт часто бажає уточнити матеріал, розміри, терміни виготовлення або замовити виріб за власним ескізом. Використання інтернет-магазину дає змогу автоматизувати значну частину цих процесів, зменшити кількість ручної роботи та зробити діяльність майстерні більш організованою.

Метою кваліфікаційної роботи є створення web-сайту інтернет-магазину «Лісова Майстерня», який забезпечує перегляд каталогу виробів, оформлення замовлень, подання індивідуальних заявок, реєстрацію та авторизацію користувачів, роботу особистого кабінету клієнта, а також адміністрування товарів, користувачів і замовлень.

Для досягнення поставленої мети необхідно виконати такі завдання:

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

провести аналітичний огляд існуючих рішень у сфері інтернет-магазинів; сформулювати технічне завдання на розробку сайту; розробити структуру основних сторінок; спроектувати базу даних для збереження користувачів, товарів, замовлень та індивідуальних заявок; реалізувати клієнтську частину з використанням React та Vite; створити серверну частину на основі Spring Boot; розробити адміністративну частину для керування товарами, користувачами та замовленнями; провести тестування основних функцій сайту; підготувати інструкції з розміщення, обслуговування та підтримки програмного продукту.

Об'єктом дослідження є процес створення та функціонування інтернет-магазину крафтових виробів.

Предметом дослідження є програмні засоби, методи та технології розробки web-сайту для продажу виробів ручної роботи, оформлення замовлень і керування даними магазину.

Практичне значення роботи полягає у створенні програмного продукту, який може бути використаний як навчальний приклад повноцінного інтернет-магазину або як основа для подальшого розвитку реального сайту майстерні. Розроблений web-сайт містить основні можливості, необхідні для роботи невеликого онлайн-магазину: каталог товарів, кошик, оформлення замовлення, індивідуальні заявки, особистий кабінет клієнта та адміністративну панель.

Під час розробки використано сучасний технологічний стек. Клієнтська частина реалізована з використанням React, React Router, Vite, SCSS та бібліотеки axios для взаємодії з API. Серверна частина створена на основі Spring Boot із застосуванням Spring Security, Spring Data JPA та Flyway. Для збереження даних використано PostgreSQL.

Структура кваліфікаційної роботи складається із загального розділу, розділу розробки технічного та робочого проєкту, спеціального розділу, економічного розділу, розділу з охорони праці, техніки безпеки та екологічних вимог, а також висновків[1]. У роботі послідовно розглянуто аналіз предметної області, проєктування структури сайту, реалізацію програмної частини, тестування та питання подальшого розміщення й підтримки web-сайту.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

1 ЗАГАЛЬНИЙ РОЗДІЛ

1.1 Аналітичний огляд існуючих рішень

У сучасних умовах розвитку електронної комерції інтернет-магазини стали одним із найпоширеніших інструментів продажу товарів і взаємодії з клієнтами. Якщо раніше web-сайт здебільшого виконував інформаційну функцію, то сьогодні він є повноцінною частиною бізнес-процесів підприємства. Через сайт користувач може переглядати асортимент, отримувати детальну інформацію про товар, оформлювати замовлення, залишати контактні дані та відстежувати стан покупки.

Особливо важливим використання інтернет-магазину є для малого бізнесу та виробників крафтової продукції. Такі виробники часто не мають великих торгових площ або розгалуженої мережі фізичних магазинів, тому онлайн-присутність стає для них основним способом залучення клієнтів. Для майстерень, що виготовляють вироби ручної роботи, сайт виконує не лише комерційну, а й презентаційну функцію, оскільки дозволяє показати стиль майстра, якість матеріалів, приклади готових робіт та можливість індивідуального замовлення.

Для продажу крафтових виробів можуть використовуватися різні цифрові рішення: сторінки в соціальних мережах, маркетплейси, конструктори сайтів, CRM-системи та власні інтернет-магазини. Кожен із цих варіантів має певні переваги, проте не всі вони повністю відповідають потребам майстерні, яка працює не лише з готовими товарами, а й з індивідуальними замовленнями.

Одним із найпростіших способів просування виробів є використання соціальних мереж. Instagram, Facebook, TikTok та інші платформи дозволяють швидко створити сторінку майстерні, публікувати фотографії виробів, відео процесу виготовлення, відгуки клієнтів та новини. Перевагою такого підходу є простота запуску, відсутність потреби у складній технічній підтримці та доступ до великої аудиторії.

Проте соціальні мережі мають суттєві обмеження. Вони не забезпечують повноцінної структури інтернет-магазину: каталог товарів часто представлений у

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

вигляді дописів або сторіс, замовлення приймаються через особисті повідомлення, а облік клієнтів і заявок ведеться вручну. У разі збільшення кількості замовлень такий спосіб роботи стає незручним, оскільки складно відстежувати статуси, контролювати оплату, залишки товарів і контактні дані покупців.

Іншим поширеним рішенням є використання маркетплейсів. Вони надають продавцю готову торговельну платформу, де можна розмістити товари, додати опис, ціну, фотографії та приймати замовлення. Перевагою такого підходу є доступ до вже сформованої аудиторії покупців, наявність пошуку, фільтрів і готового механізму продажу. Однак маркетплейси не завжди зручні для крафтових виробів, оскільки продавець обмежений правилами платформи, стандартним виглядом сторінок, комісіями та вимогами до оформлення товарів.

Для майстерні, яка виготовляє вироби за індивідуальними параметрами, важливо мати можливість гнучко збирати інформацію від клієнта: розміри, матеріал, побажання, посилання на приклади та коментарі щодо майбутнього виробу. На маркетплейсах така логіка часто реалізована обмежено або взагалі відсутня, тому вони більше підходять для продажу типових товарів, ніж для повноцінної роботи з індивідуальними заявками.

Ще одним варіантом є створення сайту за допомогою конструкторів. Такі сервіси дозволяють швидко зібрати сторінку або невеликий магазин без глибоких знань програмування. Вони зручні для простих сайтів-візиток або невеликих каталогів, однак мають обмежені можливості для реалізації власної бізнес-логіки. Якщо потрібно створити окремі ролі користувачів, особистий кабінет, історію замовлень, адміністративну панель або індивідуальні заявки, стандартних можливостей конструктора може бути недостатньо.

CRM-системи[2] також можуть використовуватися у роботі майстерні для обліку клієнтів, заявок, замовлень, оплат і комунікації з покупцями. Вони зручні для внутрішньої роботи менеджера або адміністратора, оскільки дозволяють бачити, на якому етапі перебуває замовлення та хто відповідальний за його виконання. Проте CRM-система не є повноцінною заміною інтернет-магазину,

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

адже вона переважно орієнтована на внутрішні процеси підприємства, а не на публічне представлення товарів для покупця.

Найбільш гнучким рішенням є розробка власного інтернет-магазину. У такому випадку структура сайту, дизайн, логіка роботи, база даних і адміністративні функції створюються відповідно до потреб конкретного проєкту. Власний сайт дозволяє реалізувати каталог товарів, кошик, оформлення замовлення, особистий кабінет, систему ролей, індивідуальні заявки, зміну статусів, коментарі та облік оплат.

Для більш наочного порівняння існуючих рішень доцільно подати їх характеристику у вигляді таблиці 1.1.

Таблиця 1.1 - Порівняльна характеристика цифрових рішень для продажу крафтових виробів

Тип рішення	Переваги	Недоліки	Доцільність
Соціальні мережі	Швидкий запуск, велика аудиторія	Ручне приймання замовлень	Для реклами та просування
Маркетплейси	Готова платформа, пошук товарів	Комісії та обмеження	Для продажу типових товарів
Конструктори сайтів	Обмежений функціонал	Обмежений функціонал	Для невеликих проєктів
CRM-системи	Облік клієнтів і замовлень	Не замінюють магазин	Як допоміжний інструмент
Власний інтернет-магазин	Повний контроль і гнучкість	Потребує розробки	Найкращий варіант для проєкту

Як видно з таблиці 1.1, кожне з розглянутих цифрових рішень має певні переваги, проте не всі вони повністю відповідають потребам майстерні, яка працює з крафтовими виробами та індивідуальними замовленнями. Соціальні мережі доцільно використовувати для реклами та залучення клієнтів, маркетплейси - для додаткового каналу продажів, а CRM-системи - для внутрішнього обліку. Однак найбільш повно вимогам проєкту відповідає саме власний інтернет-магазин, оскільки він дозволяє реалізувати каталог товарів,

кошик, особистий кабінет, індивідуальні заявки та адміністративну частину.

Для інтернет-магазину крафтових виробів «Лісова Майстерня» саме власний web-сайт є найбільш доцільним рішенням. Це пояснюється тим, що майстерня працює не лише з готовими товарами, а й з виробами під замовлення. Користувач повинен мати можливість переглянути готовий каталог, оформити покупку, а також подати індивідуальну заявку із зазначенням розмірів, матеріалу, опису та посилань на приклади.

Розробка власного сайту також дає можливість обрати сучасний технологічний стек. У даному проєкті клієнтська частина реалізується за допомогою React[3] та Vite[4], серверна частина - за допомогою Spring Boot[5], а для збереження даних використовується PostgreSQL[6]. Такий підхід дозволяє розділити frontend і backend, забезпечити зручну підтримку проєкту та можливість подальшого розвитку.

Додатковою перевагою власного web-сайту є можливість подальшого розвитку проєкту. У майбутньому до інтернет-магазину можна додати онлайн-оплату, модуль доставки, відгуки покупців, систему знижок, фільтри за матеріалами чи категоріями та розширену статистику для адміністратора.

Для майстерні крафтових виробів важливо зберігати власний стиль подання продукції, оскільки вироби ручної роботи потребують якісного візуального представлення, детального опису матеріалів і можливості узгодження індивідуальних побажань клієнта. Саме власний інтернет-магазин дозволяє поєднати комерційні функції з презентацією майстерні.

Отже, аналіз існуючих рішень показує, що соціальні мережі, маркетплейси, конструктори сайтів і CRM-системи можуть частково вирішувати окремі завдання, проте не забезпечують повного набору необхідних функцій. Для реалізації каталогу, кошика, індивідуальних заявок, особистого кабінету та адміністративної частини найкраще підходить розробка власного web-сайту «Лісова Майстерня» як окремого програмного продукту.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2 Технічне завдання

Технічне завдання є одним із основних етапів підготовки до розробки програмного продукту, оскільки саме в ньому визначаються призначення майбутньої системи, її функціональні можливості, вимоги до структури, програмного забезпечення, документації, тестування та подальшого приймання. Для web-сайту інтернет-магазину крафтових виробів «Лісова Майстерня» технічне завдання формує загальне уявлення про те, яким має бути програмний продукт та які результати мають бути отримані після завершення розробки.

Розроблюваний web-сайт призначений для представлення та продажу крафтових виробів ручної роботи, які можуть бути як готовими товарами, так і виготовлятися за індивідуальним замовленням клієнта. Основною особливістю такого магазину є необхідність поєднання звичайного каталогу товарів із можливістю подання індивідуальних заявок, де користувач може вказати власні побажання щодо виробу, матеріалів, розмірів, дизайну та інших параметрів.

У межах технічного завдання передбачається створення повноцінного web-сайту з клієнтською та адміністративною частинами. Клієнтська частина повинна забезпечувати зручну взаємодію покупця з магазином: перегляд товарів, пошук необхідної продукції, додавання товарів до кошика, оформлення замовлення, реєстрацію та авторизацію, перегляд власних замовлень і створення індивідуальних заявок. Адміністративна частина має забезпечити керування товарами, користувачами, замовленнями та заявками.

Розробка програмного продукту виконується з урахуванням сучасних підходів до побудови web-додатків. Клієнтська частина реалізовується з використанням React та Vite, що дає змогу створити швидкий, зручний і гнучкий інтерфейс користувача. Серверна частина реалізовується на основі Spring Boot, що забезпечує надійну обробку запитів, роботу з базою даних, авторизацію користувачів і взаємодію між frontend та backend. Для збереження інформації використовується PostgreSQL, а для організації доступу до даних застосовується Spring Data JPA[7].

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

1.2.1 Найменування та область застосування

Найменування програмного продукту - web-сайт інтернет-магазину крафтових виробів «Лісова Майстерня».

Web-сайт «Лісова Майстерня» призначений для використання у сфері електронної комерції, зокрема для продажу виробів ручної роботи через мережу Інтернет. Основним напрямом діяльності магазину є представлення та реалізація дерев'яних крафтових виробів, які можуть бути виготовлені за готовими зразками або за індивідуальними побажаннями клієнта.

Область застосування web-сайту охоплює перегляд асортименту, вибір товарів, оформлення замовлень, надсилання індивідуальних заявок, реєстрацію, авторизацію, перегляд історії замовлень і керування інформацією з боку адміністратора. Для користувача сайт є інструментом пошуку та замовлення виробів, а для адміністратора - засобом обліку товарів, заявок, замовлень і клієнтів.

Користувачами web-сайту можуть бути покупці, які переглядають каталог і оформлюють замовлення, а також адміністратор або менеджер, які виконують обробку заявок і керування даними магазину. Для покупця система спрощує вибір виробу, а для працівника майстерні - організовує продажі та контроль інформації про товари.

Застосування такого web-сайту є доцільним для невеликого бізнесу, який хоче мати власний канал продажу в Інтернеті. На відміну від соціальних мереж, сайт дозволяє впорядкувати каталог, зберігати замовлення в базі даних, розмежовувати права доступу та зручно працювати з клієнтами.

Програмний продукт може бути використаний як основа для реального інтернет-магазину або як навчальний приклад web-додатку з розділенням клієнтської та серверної частини. Його структура дозволяє надалі розширювати функціональність, додавати нові категорії товарів, онлайн-оплату, повідомлення, відгуки, доставку та інші можливості.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2.2 Призначення розробки

Призначенням розробки є створення web-сайту, який забезпечує автоматизацію основних процесів роботи інтернет-магазину крафтових виробів. Сайт повинен надати користувачам можливість зручно переглядати товари, отримувати інформацію про їх характеристики, додавати товари до кошика та оформлювати замовлення без необхідності звертатися до продавця через сторонні засоби комунікації.

Для майстерні «Лісова Майстерня» програмний продукт має виконувати не лише функцію онлайн-каталогу, а й роль інструмента для організації роботи з клієнтами. Оскільки крафтові вироби часто виготовляються індивідуально, важливо передбачити можливість подання заявки з описом бажаного виробу. Така заявка повинна містити основні побажання клієнта, наприклад опис виробу, розміри, матеріал, коментарі або інші дані, необхідні для попереднього узгодження замовлення.

Також призначенням сайту є спрощення роботи адміністратора. Адміністратор повинен мати можливість додавати нові товари, редагувати інформацію про них, змінювати ціни, керувати наявністю, переглядати замовлення користувачів, змінювати їх статуси та обробляти індивідуальні заявки. Завдяки цьому зменшується кількість ручної роботи, пов'язаної з обліком замовлень, а інформація зберігається в єдиній базі даних.

Розробка web-сайту також має навчальне призначення. У межах кваліфікаційної роботи проєкт демонструє використання сучасних технологій web-розробки, принципів клієнт-серверної архітектури, роботи з базою даних, авторизації користувачів, створення REST API та побудови адміністративної панелі.

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
						16
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2.3 Вимоги до функціоналу web-сайту

Web-сайт інтернет-магазину «Лісова Майстерня» повинен забезпечувати набір функцій, необхідних для зручної роботи користувачів, адміністратора та коректної взаємодії між клієнтською і серверною частинами. Оскільки проєкт орієнтований на продаж крафтових виробів, функціонал сайту має враховувати не лише стандартне оформлення замовлень, а й можливість подання індивідуальних заявок на виготовлення виробів за параметрами клієнта.

Для звичайного користувача основними можливостями мають бути перегляд головної сторінки, ознайомлення з інформацією про магазин, перегляд каталогу товарів, відкриття детальної сторінки товару, додавання товару до кошика, оформлення замовлення, реєстрація та авторизація в системі. Користувач повинен мати змогу швидко знайти потрібний товар, переглянути його опис, ціну, зображення та інші характеристики, після чого додати товар до кошика або перейти до оформлення покупки.

Каталог товарів повинен містити інформацію про крафтові вироби, зокрема назву, опис, ціну, зображення, категорію та основні характеристики. Товари мають відображатися у зручному для користувача вигляді, щоб він міг переглядати асортимент, порівнювати вироби між собою та переходити до детальної інформації про конкретний товар. Для крафтової продукції важливим є якісне візуальне представлення, оскільки покупець часто приймає рішення на основі зовнішнього вигляду виробу.

Окремою функцією web-сайту має бути кошик. Користувач повинен мати можливість додати товар до кошика, переглянути список вибраних товарів, змінити кількість або видалити товар перед оформленням замовлення. Після підтвердження замовлення інформація повинна передаватися на сервер і зберігатися в базі даних. Користувач, який авторизувався в системі, повинен мати можливість переглядати історію власних замовлень і бачити їх поточний статус.

Важливою вимогою є реалізація індивідуальних заявок. Оскільки магазин орієнтований на крафтові вироби, користувач повинен мати можливість подати

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

заявку на виготовлення виробу за власними параметрами. У такій заявці можуть зазначатися опис бажаного виробу, розміри, матеріал, побажання щодо оформлення, коментарі та контактна інформація. Це дозволяє зробити сайт більш придатним саме для майстерні, а не лише для продажу готових товарів.

Для захисту даних і розмежування доступу необхідно реалізувати систему реєстрації та авторизації користувачів. Авторизація повинна здійснюватися за допомогою JWT, що дозволяє безпечно передавати дані між клієнтською і серверною частинами. У системі мають бути передбачені ролі користувачів, зокрема звичайний користувач і адміністратор. Звичайний користувач має доступ до клієнтської частини сайту, а адміністратор - до функцій керування магазином.

Адміністративна частина повинна забезпечувати керування товарами, замовленнями, користувачами та індивідуальними заявками. Адміністратор повинен мати можливість додавати, редагувати й видаляти товари, переглядати список замовлень, змінювати їх статуси, переглядати дані користувачів і працювати із заявками на індивідуальне виготовлення. Це дозволяє підтримувати актуальність інформації на сайті та організувати роботу з клієнтами в єдиній системі.

Таблиця 1.2 - Функціональні вимоги до web-сайту «Лісова Майстерня»

Функція	Користувач системи
Перегляд каталогу та товарів	Користувач / Адміністратор
Робота з кошиком і оформлення замовлення	Користувач / Адміністратор
Подання індивідуальної заявки	Користувач / Адміністратор
Реєстрація, авторизація, особистий кабінет	Користувач / Адміністратор
Керування товарами та замовленнями	Адміністратор
Керування користувачами та заявками	Адміністратор

Як видно з таблиці 1.2, функціонал web-сайту поділяється на дві основні частини: клієнтську та адміністративну. Клієнтська частина орієнтована на покупця і забезпечує перегляд товарів, оформлення замовлень, роботу з кошиком,

авторизацію та подання індивідуальних заявок. Адміністративна частина призначена для керування даними магазину та обробки замовлень, що дозволяє підтримувати сайт в актуальному стані.

Також web-сайт повинен забезпечувати коректну взаємодію між клієнтською та серверною частинами. Усі дії користувача, пов'язані з оформленням замовлення, авторизацією або надсиланням заявки, повинні супроводжуватися передачею даних через API та їх подальшим збереженням у базі даних PostgreSQL. У разі виникнення помилок система повинна повідомляти користувача про причину проблеми та пропонувати повторити дію.

Важливою вимогою є адаптивність web-сайту. Інтерфейс повинен коректно відображатися на різних типах пристроїв: персональних комп'ютерах, ноутбуках, планшетах і смартфонах. Елементи навігації, форми, кнопки та картки товарів повинні залишатися зручними для використання незалежно від розміру екрана. Це особливо важливо для сучасних користувачів, які часто здійснюють покупки через мобільні пристрої.

Сайт також повинен забезпечувати достатню швидкість роботи та зручність навігації. Перехід між сторінками має виконуватися без значних затримок, а структура меню повинна бути зрозумілою навіть для нового користувача. Для покращення взаємодії із сайтом доцільно використовувати повідомлення про успішне виконання дій, попередження про помилки та візуальне підтвердження виконаних операцій.

Окремо необхідно врахувати вимоги до збереження та обробки даних. Інформація про користувачів, товари, замовлення та індивідуальні заявки повинна зберігатися в базі даних PostgreSQL. Доступ до цих даних має здійснюватися через серверну частину, що дозволяє контролювати правильність запитів, виконувати перевірку прав доступу та забезпечувати цілісність інформації.

Таким чином, функціонал web-сайту «Лісова Майстерня» повинен забезпечувати повноцінну роботу інтернет-магазину, підтримувати взаємодію між користувачем і адміністратором, а також створювати зручні умови для перегляду товарів, оформлення замовлень і подання індивідуальних заявок.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

1.2.4 Вимоги до програмної документації

Програмна документація повинна містити відомості, необхідні для розуміння структури, призначення, функціональних можливостей і принципів роботи web-сайту «Лісова Майстерня». Вона має бути достатньою для ознайомлення з проєктом, його запуску, перевірки працездатності, подальшого обслуговування та можливого розширення функціоналу.

До складу програмної документації мають входити опис предметної області, технічне завдання, опис структури сайту, характеристика основних web-сторінок, схема бази даних, опис клієнтської та серверної частини, перелік основних функцій, інструкція із запуску й розміщення сайту, а також результати тестування. Документація повинна пояснювати, які технології використані в проєкті, як організована взаємодія між frontend і backend, які дані зберігаються в базі та які функції доступні різним ролям користувачів.

Окрему увагу потрібно приділити графічним матеріалам. У звіті доцільно подати структурну схему сайту, схему шаблону web-сторінки, ER-діаграму бази даних, блок-схеми окремих процедур або схеми взаємодії користувача із системою. Такі матеріали дозволяють наочно показати логіку побудови програмного продукту та підтвердити виконання основних етапів проєктування.

Також програмна документація повинна бути корисною для подальшого супроводу web-сайту. У разі внесення змін до функціоналу, структури бази даних або інтерфейсу вона має допомагати швидко зрозуміти логіку роботи системи та призначення її основних компонентів. Це особливо важливо для проєкту, який може розширюватися після завершення кваліфікаційної роботи.

Документація повинна бути оформлена відповідно до вимог до кваліфікаційної роботи. Текст має бути структурованим, послідовним і зрозумілим, із використанням нумерованих розділів, підрозділів, таблиць, рисунків і пояснень. Усі рисунки та таблиці повинні мати назви, нумерацію та посилання в тексті.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2.5 Техніко-економічні показники

Техніко-економічні показники web-сайту «Лісова Майстерня» характеризують доцільність його розробки, очікувані витрати, ефективність використання та можливість подальшого впровадження. Оскільки програмний продукт розробляється в межах кваліфікаційної роботи, основними витратами є витрати часу розробника, використання комп'ютерної техніки, електроенергії, програмних засобів і ресурсів для тестування.

Розробка власного інтернет-магазину є економічно доцільною для невеликої майстерні, оскільки дозволяє зменшити залежність від сторонніх платформ, маркетплейсів і соціальних мереж. Власний сайт не потребує постійної сплати комісій за кожне замовлення, дає можливість самостійно керувати структурою каталогу, дизайном, функціоналом і даними клієнтів. У перспективі це може зменшити витрати на обслуговування продажів і підвищити ефективність роботи з покупцями.

До технічних показників програмного продукту можна віднести використання клієнт-серверної архітектури, розділення frontend і backend, застосування бази даних PostgreSQL, підтримку авторизації користувачів, можливість адміністрування товарів і замовлень, а також подальше масштабування системи. Такі характеристики роблять сайт придатним для розвитку й розширення.

Економічні показники детально розглядаються в економічному розділі кваліфікаційної роботи. Там визначаються стадії технологічного процесу, витрати на оплату праці, матеріальні витрати, витрати на електроенергію, амортизаційні відрахування, накладні витрати, собівартість, ціна науково-дослідної роботи, економічна ефективність і термін окупності капітальних вкладень.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						21
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2.6 Стадії та етапи розробки

Розробка web-сайту інтернет-магазину крафтових виробів «Лісова Майстерня» виконується поетапно. Такий підхід дозволяє послідовно перейти від аналізу предметної області до створення готового програмного продукту, його тестування та підготовки до використання. Поділ процесу розробки на окремі етапи дає змогу краще організувати роботу, визначити основні завдання для кожної стадії та контролювати відповідність результатів поставленим вимогам.

Першим етапом є аналіз предметної області та існуючих рішень. На цьому етапі розглядаються особливості роботи інтернет-магазинів, соціальних мереж, маркетплейсів, конструкторів сайтів і CRM-систем. Визначаються їхні переваги та недоліки для майстерні, що займається виготовленням і продажем крафтових виробів. Особлива увага приділяється тому, наскільки ці рішення підходять для роботи з індивідуальними замовленнями, адже для «Лісової Майстерні» важливо не лише продавати готові товари, а й приймати заявки на виготовлення виробів за параметрами клієнта. На основі проведеного аналізу обґрунтовується доцільність розробки власного web-сайту.

Другим етапом є формування технічного завдання. На цьому етапі визначаються найменування та область застосування програмного продукту, його призначення, основні функціональні можливості, вимоги до програмної документації, техніко-економічні показники, а також порядок тестування та прийому. Технічне завдання є основою для подальшого проєктування й реалізації сайту, оскільки саме воно визначає, які функції повинні бути реалізовані та яким вимогам має відповідати готовий програмний продукт.

Третім етапом є проєктування структури web-сайту. На цьому етапі визначаються основні сторінки сайту, логіка переходів між ними, структура клієнтської та адміністративної частини, ролі користувачів і загальний принцип взаємодії з базою даних. До основних сторінок клієнтської частини належать головна сторінка, каталог товарів, сторінка детального перегляду товару, кошик, сторінка оформлення замовлення, сторінки реєстрації та авторизації, особистий

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

кабінет користувача і форма індивідуальної заявки. Для адміністративної частини передбачаються сторінки керування товарами, замовленнями, користувачами та заявками.

На цьому ж етапі проєктується структура бази даних. Визначаються основні сутності системи, зокрема користувачі, товари, категорії, замовлення, позиції замовлення та індивідуальні заявки. Для кожної сутності визначаються основні поля, зв'язки з іншими таблицями та призначення в межах роботи сайту. Правильно спроектована база даних є важливою умовою стабільної роботи інтернет-магазину, оскільки саме в ній зберігається основна інформація про користувачів, товари, замовлення та заявки.

Четвертим етапом є реалізація клієнтської частини web-сайту. На цьому етапі створюються сторінки сайту, компоненти інтерфейсу, маршрутизація, форми, кошик, особистий кабінет і механізм взаємодії з серверною частиною. Для розробки клієнтської частини використовується React, Vite, React Router, SCSS та бібліотека axios[8]. React дозволяє створювати інтерфейс на основі окремих компонентів, Vite забезпечує швидкий запуск і збірку проєкту, React Router відповідає за переходи між сторінками, SCSS використовується для стилізації, а axios - для надсилання HTTP-запитів до API.

П'ятим етапом є реалізація серверної частини. Вона створюється з використанням Spring Boot і відповідає за обробку запитів, роботу з базою даних, авторизацію користувачів, перевірку прав доступу та збереження інформації. У серверній частині створюються контролери, сервіси, репозиторії, моделі даних і механізм взаємодії з PostgreSQL. Для доступу до даних використовується Spring Data JPA, а для організації безпеки - Spring Security[9] та JWT. Завдяки цьому звичайні користувачі мають доступ лише до клієнтських функцій, а адміністративні можливості залишаються доступними тільки для користувачів із відповідною роллю.

Шостим етапом є інтеграція клієнтської та серверної частин. На цьому етапі налаштовується обмін даними між frontend і backend за допомогою HTTP-запитів. Перевіряється отримання списку товарів, передача даних реєстрації та

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						23
Зм.	Арк.	№ докум.	Підпис	Дата		

авторизації, оформлення замовлення, подання індивідуальної заявки, отримання історії замовлень і робота адміністративних функцій. Інтеграція є важливим етапом, оскільки саме вона забезпечує повноцінну роботу сайту як єдиної системи.

Сьомим етапом є тестування web-сайту. Перевіряється робота основних функцій: реєстрація, авторизація, перегляд каталогу, відкриття сторінки товару, додавання товарів до кошика, оформлення замовлення, подання індивідуальної заявки, перегляд замовлень і робота адміністративної панелі. Окремо перевіряється правильність збереження даних у базі, обробка помилок, захист адміністративних сторінок і коректність роботи API-запитів.

Восьмим етапом є виправлення виявлених помилок і підготовка програмного продукту до прийому. Після тестування аналізуються результати перевірки, усуваються недоліки, уточнюються окремі елементи інтерфейсу та перевіряється стабільність роботи сайту. На завершальному етапі готується програмна документація, описуються результати виконаної роботи, порядок запуску, обслуговування та можливості подальшого розвитку web-сайту.

Важливою умовою виконання всіх етапів є їх послідовність, оскільки кожна наступна стадія спирається на результати попередньої. Наприклад, програмування клієнтської та серверної частин неможливо якісно виконати без попереднього проєктування структури сайту та бази даних. Так само тестування має проводитися після інтеграції основних модулів, щоб перевірити роботу web-сайту як єдиної системи.

Отже, розробка інтернет-магазину «Лісова Майстерня» передбачає послідовне виконання взаємопов'язаних етапів: аналіз предметної області, формування технічного завдання, проєктування структури сайту і бази даних, створення клієнтської частини, реалізацію серверної частини, інтеграцію, тестування та підготовку до прийому. Такий підхід дозволяє отримати цілісний програмний продукт, який відповідає поставленим вимогам і може бути використаний як основа для подальшого розвитку інтернет-магазину.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						24
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2.7 Порядок тестування та прийому

Тестування web-сайту «Лісова Майстерня» виконується з метою перевірки правильності роботи основних функцій, відповідності програмного продукту технічному завданню та готовності сайту до використання.

На першому етапі тестування перевіряється коректність відображення сторінок сайту. Головна сторінка, каталог товарів, сторінка товару, кошик, сторінка оформлення замовлення, сторінки авторизації та реєстрації повинні відкриватися без помилок, мати зрозумілий інтерфейс і коректно відображатися в браузері. Також перевіряється робота навігації між сторінками.

На другому етапі перевіряються функції користувача. Необхідно протестувати реєстрацію нового облікового запису, авторизацію, вихід із системи, додавання товарів до кошика, зміну кількості товарів, видалення товарів із кошика, оформлення замовлення та перегляд історії замовлень. Окремо перевіряється подання індивідуальної заявки.

На третьому етапі перевіряється робота адміністративної частини. Адміністратор повинен мати доступ до функцій керування товарами, замовленнями, користувачами та заявками. Перевіряється додавання нового товару, редагування інформації про товар, видалення товару, перегляд списку замовлень, зміна статусу замовлення та перегляд індивідуальних заявок.

На четвертому етапі виконується перевірка взаємодії клієнтської та серверної частин. Перевіряється правильність надсилання HTTP-запитів, отримання відповідей від API, збереження даних у PostgreSQL і обробка помилок. Також необхідно перевірити, що користувач без прав адміністратора не може отримати доступ до адміністративних функцій.

Прийом програмного продукту здійснюється після успішного виконання основних тестових сценаріїв. Web-сайт вважається придатним до використання, якщо основні сторінки відкриваються без помилок, користувач може оформити замовлення, подати індивідуальну заявку, переглянути власні замовлення, а адміністратор може керувати товарами, замовленнями та заявками.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						25
Зм.	Арк.	№ докум.	Підпис	Дата		

2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ

2.1 Розробка структури сайту і web-сторінок

Розробка структури сайту і web-сторінок є одним із основних етапів створення інтернет-магазину крафтових виробів «Лісова Майстерня». На цьому етапі визначається призначення майбутнього web-ресурсу, його основні розділи, логіка переходів між сторінками, склад функціональних можливостей, а також загальна побудова інтерфейсу користувача. Правильно спроектована структура сайту дозволяє зробити ресурс зручним, зрозумілим для користувача та придатним для подальшої реалізації.

Відповідно до технічного завдання, розроблюваний сайт належить до web-ресурсів комерційного призначення. Його основною метою є представлення та продаж крафтових виробів ручної роботи через мережу Інтернет. Web-сайт «Лісова Майстерня» призначений для майстерні, яка займається виготовленням дерев'яних виробів, продажем готової продукції та прийманням індивідуальних замовлень від клієнтів. Тому структура сайту повинна поєднувати можливості звичайного інтернет-магазину та системи подання заявок на виготовлення виробів за параметрами користувача.

Розроблюваний web-сайт не є персональним сайтом, електронною енциклопедією або сайтом дистанційного навчання. За своїм призначенням він є сайтом організації, а саме інтернет-магазином із клієнтською, менеджерською та адміністративною частинами. Клієнтська частина призначена для покупців і відвідувачів сайту, менеджерська - для роботи із замовленнями, а адміністративна - для керування товарами, користувачами та замовленнями. Такий поділ дозволяє розмежувати можливості різних ролей користувачів і забезпечити впорядковану роботу програмного продукту.

Під час аналізу технічного завдання було визначено основні завдання, які ставляться перед web-сайтом. Сайт повинен забезпечувати перегляд каталогу товарів, відкриття детальної сторінки виробу, пошук і перегляд товарів за

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						26
Зм.	Арк.	№ докум.	Підпис	Дата		

категоріями, оформлення замовлення, створення індивідуальної заявки, реєстрацію та авторизацію користувачів, роботу особистого кабінету, а також адміністрування даних. Окрему увагу потрібно приділити зручності навігації, оскільки користувач повинен швидко знаходити потрібний розділ і без складнощів виконувати основні дії.

Під час проєктування структури сайту було враховано, що користувачі можуть мати різні ролі. Неавторизований користувач може переглядати головну сторінку, каталог товарів, детальну інформацію про товари, користуватися пошуком, а також перейти до сторінки входу або реєстрації. Авторизований користувач отримує доступ до профілю, може переглядати історію власних замовлень, оформлювати покупки та подавати індивідуальні заявки. Менеджер може працювати із замовленнями, а адміністратор отримує доступ до сторінок керування товарами, користувачами та замовленнями. UML-діаграму[10] варіантів використання web-сайту «Лісова Майстерня» наведено у графічній частині кваліфікаційної роботи на 3 аркуші 2026.КВР.122.423.03.00.00 СС.

Основою структури web-сайту є головна сторінка. Вона є початковою точкою взаємодії користувача з ресурсом і виконує презентаційну функцію. На головній сторінці користувач ознайомлюється з назвою магазину, коротким описом діяльності майстерні, перевагами замовлення виробів ручної роботи та основними напрямками роботи сайту. З головної сторінки користувач може перейти до каталогу, сторінки входу, профілю користувача або службових панелей залежно від своєї ролі.

Каталог товарів є одним із головних розділів сайту. Його призначення полягає у відображенні асортименту крафтових виробів, доступних для перегляду та замовлення. У каталозі товари подаються у вигляді окремих карток. Кожна картка містить основну інформацію про виріб: назву, зображення, ціну, короткий опис і можливість перейти до детального перегляду. Також у структурі каталогу передбачено пошук, категорії товарів, кошик та оформлення замовлення.

Сторінка детального перегляду товару призначена для надання користувачу повної інформації про конкретний виріб. На ній розміщуються

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

зображення товару, назва, опис, ціна, матеріали, розміри, термін виготовлення або інші характеристики. Для крафтових виробів така сторінка має важливе значення, оскільки покупець повинен отримати достатньо інформації перед оформленням замовлення. Саме на основі детального опису та зображення користувач приймає рішення щодо покупки.

Кошик є проміжним елементом між переглядом товарів і оформленням замовлення. У кошику користувач може переглянути вибрані товари, перевірити їх кількість, ціну та загальну суму. Після цього користувач переходить до оформлення замовлення. Така послідовність відповідає логіці роботи інтернет-магазину та забезпечує зрозумілий процес покупки.

Окремою частиною структури сайту є профіль користувача. Він призначений для відображення персональної інформації та даних, пов'язаних із замовленнями. У профілі користувач може переглядати свої контактні дані, роль у системі, кількість замовлень, а також історію звичайних та індивідуальних замовлень. Наявність особистого кабінету робить сайт зручнішим для користувача, оскільки він може самостійно контролювати власну взаємодію з магазином.

Сторінка входу забезпечує авторизацію користувача в системі. Через неї користувач може увійти до вже створеного облікового запису або перейти до реєстрації. Реєстрація потрібна для створення нового користувача в системі та подальшого доступу до додаткових можливостей сайту. Після авторизації користувач отримує доступ до функцій відповідно до своєї ролі.

Адміністративна панель web-сайту призначена для керування основними даними магазину. Вона містить розділи керування товарами, замовленнями та користувачами. У розділі керування товарами адміністратор може додавати нові вироби, редагувати наявні позиції, змінювати опис, ціну, зображення, категорію, наявність і статус активності товару. Завдяки цьому каталог можна підтримувати в актуальному стані без внесення змін у програмний код.

Розділ керування користувачами використовується для перегляду та адміністрування облікових записів. Адміністратор може бачити список

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						28
Зм.	Арк.	№ докум.	Підпис	Дата		

zareestrovanih korystuvachiv, i'xni dani, roli ta pov'язanu informaciu. Це потрібно для контролю доступу до системи та правильного розподілу ролей між користувачами.

Розділ керування замовленнями є важливою частиною адміністративної та менеджерської панелі. У ньому адміністратор або менеджер може переглядати звичайні замовлення з каталогу та індивідуальні заявки користувачів. Для замовлень передбачено відображення контактних даних клієнта, складу замовлення, суми, статусу, способу оплати або іншої службової інформації. Також адміністратор або менеджер може змінювати статус замовлення, що дозволяє контролювати процес його виконання.

Після опису основних сторінок було сформовано карту сайту. Карта сайту відображає основні розділи web-ресурсу та логіку переходів між ними. Вона є важливим елементом проєктування, оскільки дозволяє наочно показати структуру майбутнього сайту, визначити зв'язки між сторінками та перевірити, чи всі необхідні функції враховані на етапі розробки.

Структурну схему web-сайту «Лісова Майстерня» наведено у графічній частині кваліфікаційної роботи на 2 аркуші 2026.KBP.122.423.03.00.00 СС. Зі схеми видно, що центральною сторінкою є головна сторінка web-сайту, з якої користувач може перейти до основних розділів: панелі адміністратора, панелі менеджера, профілю користувача, каталогу та сторінки входу. Кожен із цих розділів має власні підрозділи, що відповідають його призначенню.

Панель адміністратора містить функції керування товарами, замовленнями та користувачами. Панель менеджера призначена для керування замовленнями. Профіль користувача містить можливість перегляду профілю та історії замовлень. Каталог включає пошук, категорії товарів, кошик і оформлення замовлення. Сторінка входу пов'язана з авторизацією та реєстрацією користувача. Така структура відповідає логіці роботи інтернет-магазину та забезпечує розподіл функцій між різними частинами сайту.

Як видно з таблиці 2.1, структура сайту охоплює як клієнтські, так і службові сторінки. Клієнтські сторінки забезпечують перегляд товарів,

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
						29
Зм.	Арк.	№ докум.	Підпис	Дата		

оформлення замовлень, авторизацію та роботу з профілем. Службові сторінки призначені для керування основними даними магазину. Такий поділ дозволяє уникнути перевантаження інтерфейсу та забезпечити доступ до функцій відповідно до ролі користувача.

Таблиця 2.1 - Структура web-сайту «Лісова Майстерня»

Назва сторінки	Маршрут сторінки	Призначення
Головна сторінка	/	Ознайомлення користувача з магазином, його тематикою та основними можливостями
Каталог товарів	/products	Перегляд асортименту крафтових виробів
Детальна сторінка товару	/products/:id	Перегляд повної інформації про конкретний товар
Індивідуальне замовлення	/custom-order	Оформлення заявки на виготовлення виробу за параметрами користувача
Профіль користувача	/profile	Перегляд особистих даних і замовлень користувача
Авторизація	/login	Вхід користувача до системи
Реєстрація	/register	Створення нового облікового запису
Відновлення пароля	/forgot-password, /reset-password	Відновлення доступу до облікового запису
Керування товарами	/admin/products	Додавання, редагування та видалення товарів
Керування користувачами	/admin/users	Перегляд і керування зареєстрованими користувачами
Керування замовленнями	/admin/orders	Перегляд і обробка звичайних та індивідуальних замовлень

Під час проектування інтерфейсу було враховано необхідність створення єдиного шаблону сторінок. У web-сайті використовується спільна структура інтерфейсу, що містить навігаційну область, основну частину сторінки та службові елементи. Такий підхід дозволяє забезпечити однаковий стиль оформлення всіх сторінок і зробити навігацію зрозумілою для користувача.

На рисунку 2.1 подано схему[11] шаблону web-сторінки сайту «Лісова Майстерня». Схема показує загальне розміщення основних елементів інтерфейсу: навігаційної панелі, основної області вмісту, заголовка сторінки, блоків із даними,

форм, таблиць, кнопок дій та системних повідомлень. Використання єдиного шаблону дозволяє забезпечити узгодженість сторінок і спростити подальшу верстку web-сайту.

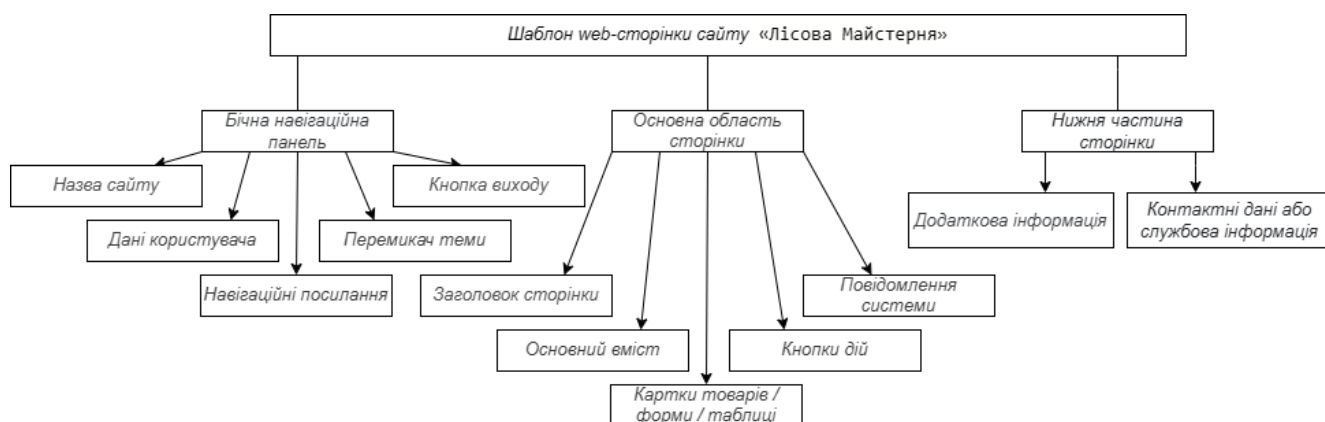


Рисунок 2.1 - Схема шаблону web-сторінки

Під час проєктування шаблону сторінки було враховано, що різні розділи сайту мають відрізнятися за змістом, але зберігати спільну логіку побудови. У каталозі основна область містить картки товарів, у профілі - персональні дані та історію замовлень, а в адміністративній панелі - таблиці й форми керування.

Проєктування дизайну виконувалося поетапно. Спочатку було визначено концепцію головної сторінки, яка має передавати тематику майстерні, натуральність і ручну роботу. Далі було спроєктовано структуру внутрішніх сторінок, які зберігають єдиний стиль оформлення, але відрізняються функціональним наповненням.

Результатом етапу є визначення призначення ресурсу, побудова карти сайту, опис основних сторінок, проєктування навігації та загального шаблону інтерфейсу. Web-сайт «Лісова Майстерня» складається з клієнтської частини для покупців, менеджерської частини для роботи із замовленнями та адміністративної частини для керування магазином.

Отже, у межах підрозділу було спроєктовано структуру web-сайту, визначено основні розділи, сторінки, ролі користувачів і принципи навігації. Запропонована структура відповідає технічному завданню та є основою для подальшої верстки, програмування і тестування.

2.2 Створення та верстка сторінок сайту

Після розробки структури web-сайту та визначення основних сторінок було виконано створення й верстку інтерфейсу інтернет-магазину крафтових виробів «Лісова Майстерня». На цьому етапі формувалася зовнішній вигляд сторінок, розміщувалися основні елементи інтерфейсу, створювалися шаблони сторінок, виконувалася стилізація блоків і налаштовувалися інтерактивні елементи.

Створення web-сторінок виконувалося у середовищі Visual Studio Code. Для клієнтської частини використовувалися React та Vite. Сторінки сайту формувалися не як окремі статичні HTML-файли, а як React-компоненти, що під час роботи додатку відображають відповідну HTML-структуру[12] в браузері. Такий підхід дозволив поділити інтерфейс на окремі частини, повторно використовувати компоненти та спростити подальшу підтримку проєкту.

Для переходів між сторінками використовується бібліотека React Router. Маршрутизація налаштована у файлі client-vite/src/App.jsx, фрагмент коду якого наведено в додатку А. У ньому визначено основні сторінки web-сайту та умови доступу до окремих розділів. Через маршрути відкриваються головна сторінка, каталог товарів, сторінка окремого товару, сторінки авторизації, реєстрації, профілю, індивідуального замовлення та адміністративні сторінки.

Верстка сторінок виконувалася з використанням компонентного підходу. Інтерфейс було поділено на навігаційну панель, область основного вмісту, картки товарів, форми, кнопки, таблиці, модальні вікна та службові повідомлення. Завдяки цьому вдалося забезпечити єдину структуру сайту та однаковий стиль оформлення на всіх сторінках.

Для стилізації сторінок використовувався SCSS, що дозволило зручно організувати стилі, створити окремі файли для різних частин інтерфейсу та використовувати вкладені правила оформлення. Під час верстки також враховувалася тематика проєкту: сайт орієнтований на продаж дерев'яних крафтових виробів, тому оформлення повинно підкреслювати натуральність,

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		32

ручну роботу та затишний стиль майстерні.

Одним із перших етапів верстки було створення загального шаблону сторінок. Він містить навігаційну область, основну частину сторінки та службові елементи. Основна область змінюється залежно від відкритої сторінки: на головній сторінці відображається презентаційна інформація, у каталозі - список товарів, у профілі - дані користувача, а в адміністративній частині - таблиці та форми керування.

Головна сторінка, рисунок 2.2 web-сайту виконує презентаційну функцію. Вона знайомить користувача з магазином «Лісова Майстерня», коротко пояснює напрям діяльності майстерні та пропонує перейти до основних розділів сайту. На головній сторінці розміщено інформаційні блоки, які підкреслюють тематику ручної роботи, натуральних матеріалів і можливості індивідуального замовлення.

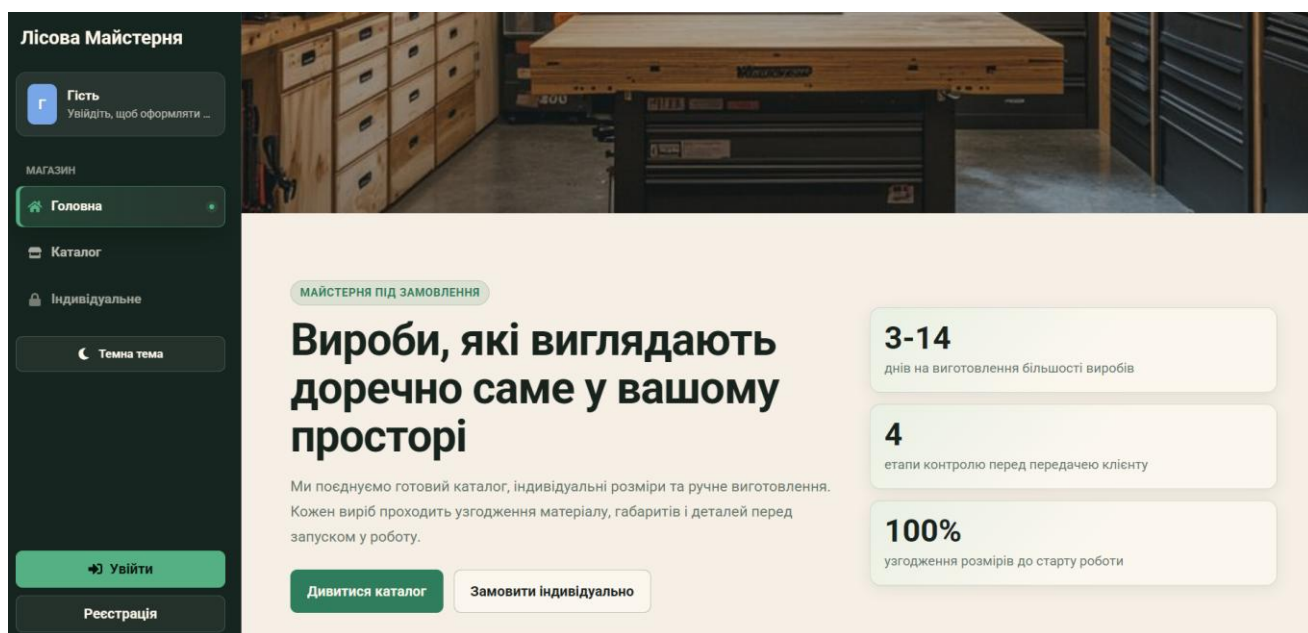


Рисунок 2.2 - Головна сторінка web-сайту «Лісова Майстерня»

Сторінка каталогу товарів створена для відображення асортименту крафтових виробів. Основним елементом цієї сторінки є картка товару. Вона містить зображення виробу, назву, короткий опис, ціну та елементи взаємодії з користувачем. Картки розміщуються у вигляді сітки, що дозволяє зручно переглядати декілька товарів одночасно, тому що так користувач зможе швидко

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
						33
Зм.	Арк.	№ докум.	Підпис	Дата		

порівняти товари між собою.

Під час верстки каталогу було враховано, що для крафтових виробів важливе візуальне представлення. Зображення товарів повинні займати помітне місце, оскільки саме вони формують перше враження про виріб. Текстова інформація в картці подається стисло, щоб не перевантажувати сторінку. Детальніший опис товару користувач може переглянути після переходу на сторінку конкретного виробу.

Для створення сторінки каталогу використовується JSX-розмітка React-компонента. У ній формуються основні елементи інтерфейсу: заголовок сторінки, вкладки каталогу, поле пошуку, вибір категорії та службові блоки керування відображенням товарів. Фрагмент JSX-розмітки сторінки каталогу наведено в додатку Б.

На рисунку 2.3 подано сторінку каталогу товарів. На ній видно принцип розміщення карток, оформлення товарів, використання зображень і загальний вигляд сторінки. Каталог є одним із ключових розділів сайту, тому його верстка повинна бути зручною та зрозумілою.

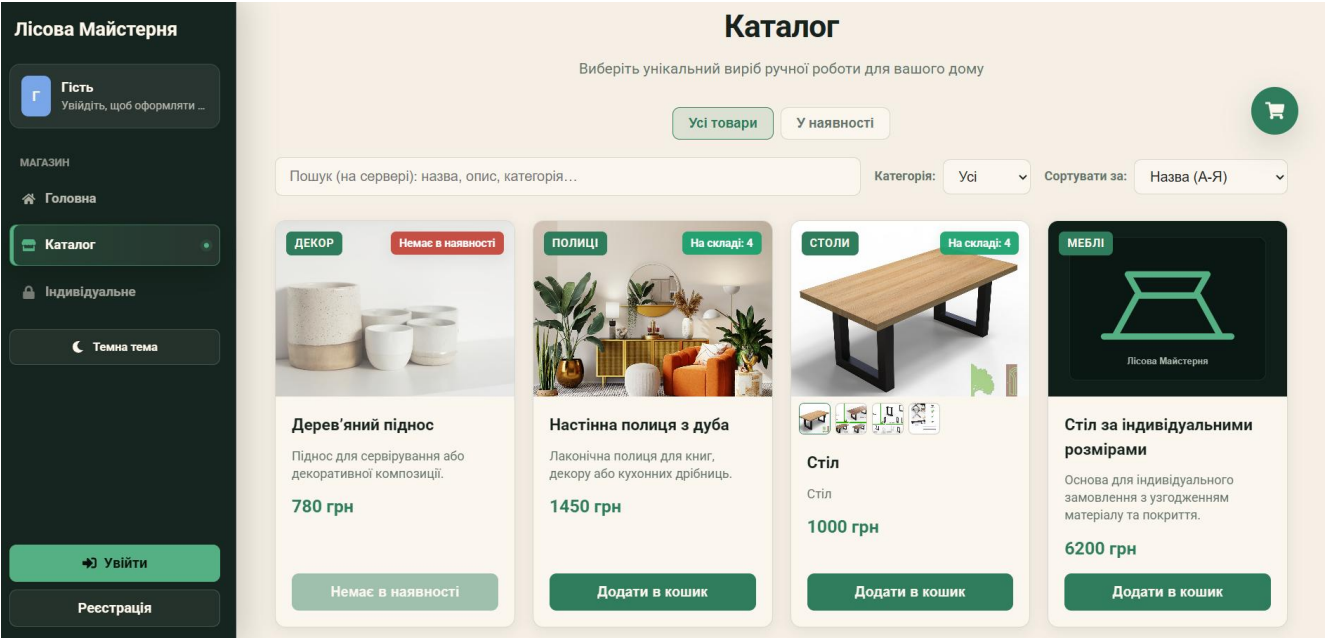


Рисунок 2.3 - Сторінка каталогу товарів

Сторінка детального перегляду товару призначена для подання повної

інформації про конкретний виріб. На ній розміщуються зображення, назва, опис, ціна та основні характеристики товару. Також передбачена можливість додати товар до кошика або перейти до оформлення замовлення. На відміну від картки товару в каталозі, детальна сторінка містить більше інформації та допомагає користувачу прийняти рішення щодо покупки.

Окремою важливою сторінкою є сторінка індивідуального замовлення. Вона створена для користувачів, які хочуть замовити виріб за власними параметрами. На цій сторінці розміщується форма, у якій користувач може вказати контактні дані, розміри виробу, матеріал, опис майбутнього замовлення та додаткові побажання. Поля форми розташовані послідовно, а службові повідомлення допомагають користувачу зрозуміти результат надсилання заявки.

На рисунку 2.4 наведено сторінку оформлення індивідуального замовлення. Вона демонструє форму для введення даних користувача та параметрів майбутнього виробу. Наявність цієї сторінки є важливою особливістю сайту «Лісова Майстерня», оскільки проєкт орієнтований не лише на продаж готових товарів, а й на приймання індивідуальних заявок.

Лісова Майстерня

Н Назар
nazar@gmail.com

МАГАЗИН

Головна

Каталог

✕ Індивідуальне

Профіль

Темна тема

Вихід

Замовити індивідуально
Опишіть ваш виріб, додайте розміри та посилання на ескіз/фото. Ми узгодимо деталі.

Контакти

Ім'я

Прізвище

Телефон

+380

Email

Параметри

Ширина (см)

Висота (см)

Глибина (см)

Матеріал

Напр. дуб, сосна...

Опис/побажання

Опишіть деталі...

Орієнтовна вартість
Попередній розрахунок для заявки

Потрібні розміри

Сума враховує розмір, глибину, матеріал, роботу та фінішну обробку. Фінальну ціну менеджер підтвердить після уточнення ескізу.

Рисунок 2.4 - Сторінка індивідуального замовлення

Сторінки авторизації та реєстрації створені для роботи з обліковими

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		35

записами. На сторінці авторизації користувач вводить дані для входу, а на сторінці реєстрації створює новий обліковий запис. У разі помилки система повідомляє про неправильні дані або незаповнені обов'язкові поля. Також передбачено відновлення пароля.

Сторінка профілю призначена для відображення персональної інформації та даних, пов'язаних із замовленнями. У профілі користувач може переглядати контактні дані, роль у системі, кількість замовлень, а також звичайні та індивідуальні заявки.

Адміністративна частина сайту орієнтована на керування даними. Для неї використовуються таблиці, кнопки дій, форми редагування та службові повідомлення, що дозволяє швидко працювати з товарами, користувачами й замовленнями.

На сторінці керування товарами відображаються назва, ціна, категорія, наявність та інші дані, а також передбачено можливість редагування товарів. Сторінка керування користувачами використовується для перегляду зареєстрованих користувачів і зміни їхніх ролей.

Сторінка керування замовленнями, рисунок 2.5, призначена для перегляду звичайних замовлень та індивідуальних заявок. Адміністратор або менеджер може переглядати контактні дані клієнта, склад замовлення, суму, статус, коментарі та змінювати стан замовлення.

Лісова Майстерня

Н Назар Гардамала
nazargardamala1234@gmail.com

МАГАЗИН

- Головна
- Каталог
- Індивідуальне
- Профіль

АДМІНІСТРУВАННЯ

- Товари
- Користувачі
- Замовлення**
- Темна тема
- Вихід

Замовлення

Створити замовлення

Нові замовлення: 4

В роботі: 4 (погоджено / в роботі / відправлено)

Оплачено: 38 900 грн

Тип: Усі

Не замовлення, телефон +380, ім'я, email, статус...

Сортувати за: Дата (новіші)

Пошук: порядковий номер (можна «№123»), повний або частковий телефон, ПІБ, email, статус або матеріал у заявках.

№	ТИП ЗАМОВЛЕННЯ	ДАТА	КЛІЄНТ	КОРОТКО	СТАТУС
9	Під замовлення	28.05.2026, 21:37:27	test testtt t@gmail.com	50x100x50 см - 3250 грн	Погоджено
8	Товари в наявності	28.05.2026, 21:36:17	Тест тест t@gmail.com	1 поз. - 2000.00 грн	Новий
7	Товари в наявності	24.05.2026, 19:57:18	Тарас Максимчук joox41851da@gmail.com	2 поз. - 2450.00 грн	Новий
6	Товари в наявності	19.05.2026, 23:29:13	Назар Гардамала nazargardamala1234@gmail.com	2 поз. - 2450.00 грн	Закрито
5	Товари в наявності	19.05.2026, 21:39:49	Олександр Ковальчик sasha67@gmail.com	1 поз. - 2000.00 грн	В роботі
4	Товари в наявності	08.05.2026, 23:49:01	гість гість	1 поз. - 1000.00 грн	Новий

Рисунок 2.5 - Сторінка керування замовленнями

На етапі верстки також було виконано скриптування інтерактивних елементів інтерфейсу. До таких елементів належать перемикання між сторінками, відкриття й закриття форм, робота з кошиком, обробка введених даних у формах, відображення повідомлень про помилки або успішні дії, а також зміна вигляду окремих елементів залежно від дій користувача. У React така логіка реалізовується через компоненти, обробники подій і стан елементів інтерфейсу.

Під час верстки сторінок було використано підхід адаптивного інтерфейсу. Це означає, що сторінки повинні коректно відображатися на різних типах пристроїв: персональних комп'ютерах, ноутбуках, планшетах і смартфонах. Елементи інтерфейсу мають змінювати своє розміщення залежно від ширини екрана, щоб користувач міг зручно працювати із сайтом незалежно від пристрою.

Окремо під час верстки було враховано однаковість оформлення елементів на різних сторінках сайту. Для кнопок, форм, карток товарів і повідомлень використовувалися спільні стилі, щоб інтерфейс виглядав цілісно. Це дозволяє користувачу швидше орієнтуватися на сайті, оскільки схожі елементи мають однакове призначення та зовнішній вигляд. Такий підхід також спрощує подальше редагування дизайну, оскільки зміни можна вносити централізовано.

У процесі верстки також враховувалися вимоги до зручності використання. Кнопки повинні мати зрозумілі підписи, форми - логічну послідовність полів, а повідомлення системи - чітко пояснювати результат дії. Це особливо важливо для інтернет-магазину, де користувач повинен без зайвих труднощів оформити замовлення або надіслати індивідуальну заявку.

Результатом виконання етапу створення та верстки сторінок сайту стали готові шаблони клієнтських і адміністративних сторінок. Було створено головну сторінку, каталог товарів, сторінку детального перегляду товару, сторінку індивідуального замовлення, сторінки авторизації, реєстрації, відновлення пароля, профіль користувача та адміністративні сторінки для керування товарами, користувачами й замовленнями. Усі ці сторінки формують єдиний web-інтерфейс інтернет-магазину «Лісова Майстерня» та є основою для подальшого програмування функціоналу сайту.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		37

2.3 Розробка структури бази даних сайту

Розробка структури бази даних є важливим етапом створення web-сайту інтернет-магазину крафтових виробів «Лісова Майстерня». База даних використовується для збереження основної інформації, необхідної для роботи сайту: даних користувачів, товарів, звичайних замовлень, індивідуальних заявок, статусів і службових номерів. Правильно спроектована база даних забезпечує цілісність інформації, зручність її обробки та стабільну роботу програмного продукту.

Під час проєктування бази даних було враховано функціональні вимоги до web-сайту. Система повинна зберігати інформацію про зареєстрованих користувачів, їх ролі, контактні дані, товари каталогу, замовлення з каталогу та індивідуальні замовлення. Також база даних повинна підтримувати роботу адміністративної частини, у якій адміністратор або менеджер може переглядати замовлення, змінювати їх статуси, працювати з товарами та користувачами.

Для збереження даних у проєкті використовується PostgreSQL. Взаємодія серверної частини з базою даних реалізована за допомогою Spring Data JPA. Структура таблиць створюється та оновлюється за допомогою міграцій Flyway, що дозволяє послідовно керувати змінами у схемі бази даних. Такий підхід є зручним для розробки, оскільки всі зміни структури зберігаються у вигляді окремих SQL-файлів і можуть бути повторно застосовані під час розгортання проєкту.

Проектування бази даних виконувалося поетапно. Спочатку було визначено основні об'єкти предметної області: користувачі, товари, замовлення, індивідуальні замовлення та службові лічильники. Далі для кожного об'єкта було визначено набір полів, необхідних для збереження даних. Після цього було встановлено зв'язки між таблицями, зокрема зв'язок користувача із його замовленнями та індивідуальними заявками.

Основними таблицями бази даних є users, products, orders, custom_orders та counters. Таблиця users призначена для збереження облікових записів

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		38

користувачів. У ній зберігаються ім'я, прізвище, електронна пошта, номер телефону, пароль, роль користувача, статус активності та службові поля для відновлення пароля. Наявність поля ролі дозволяє розмежувати доступ між звичайним користувачем, менеджером і адміністратором.

Таблиця products використовується для збереження інформації про товари каталогу. У ній містяться назва товару, матеріали, розміри, ціна, час виготовлення, посилання на фото, опис, категорія, список зображень, кількість на складі та статус активності. Ця таблиця є основою каталогу товарів, оскільки саме з неї клієнтська частина отримує дані для відображення асортименту.

Таблиця orders призначена для збереження звичайних замовлень, які користувач оформлює через каталог товарів. У ній зберігаються номер замовлення, користувач, контактні дані клієнта, місто, номер відділення Нової пошти, статус замовлення, статус оплати, спосіб оплати, сума оплати, дата оплати, примітка до оплати, список товарів у форматі JSON та історія змін замовлення.

Таблиця custom_orders використовується для індивідуальних заявок користувачів. Вона зберігає номер заявки, користувача, статус, орієнтовну ціну, інформацію про оплату, контактні дані, параметри виробу, посилання на приклади, історію оплат і часову шкалу змін. Оскільки індивідуальні замовлення можуть мати різну структуру параметрів, частина даних зберігається у форматі JSON. Це дозволяє гнучко зберігати опис виробу, розміри, матеріали, побажання клієнта та інші параметри без створення великої кількості окремих полів.

Таблиця counters є службовою. Вона використовується для збереження лічильників, які потрібні для формування послідовних номерів замовлень та іншого. Завдяки цьому кожне замовлення може отримувати зрозумілий номер, який зручно використовувати під час адміністрування та пошуку.

ER-діаграму бази даних web-сайту «Лісова Майстерня» наведено у графічній частині кваліфікаційної роботи на 1 аркуші 2026.КВР.122.423.03.00.00 БД. Діаграма відображає основні таблиці бази даних, їх поля та зв'язки між ними. Центальною таблицею для користувацьких даних є users, оскільки саме з

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						39
Зм.	Арк.	№ докум.	Підпис	Дата		

користувачем пов'язані звичайні замовлення та індивідуальні заявки. Таблиця products зберігає товари каталогу, таблиця orders - звичайні замовлення, таблиця custom_orders - індивідуальні замовлення, а таблиця counters використовується для службової нумерації.

Кожна таблиця бази даних має власне призначення та відповідає окремій частині функціоналу web-сайту. Наприклад, таблиця користувачів забезпечує збереження облікових записів і розмежування доступу, таблиця товарів використовується для формування каталогу, а таблиці замовлень відповідають за облік покупок та індивідуальних заявок. Узагальнено ці елементи подано в таблиці 2.2, де зазначено основні поля кожної таблиці та їх призначення. Окремо використовується службова таблиця лічильників, яка потрібна для формування порядкових номерів замовлень.

Таблиця 2.2 - Структура таблиць бази даних

Назва таблиці	Основні поля	Призначення
users	id, name, last_name, email, phone, password, role, is_active, created_at, updated_at	Збереження даних користувачів, ролей і стану облікових записів
products	id, name, materials, sizes, price, photo_url, description, category, stock_qty, is_active	Збереження інформації про товари каталогу
orders	id, order_number, user_id, first_name, last_name, phone, city, status, payment_status, items_json	Збереження звичайних замовлень, оформлених через каталог
custom_orders	id, order_number, user_id, status, estimated_price, contact_json, params_json, reference_urls_json	Збереження індивідуальних заявок на виготовлення виробів
counters	counter_key, seq	Збереження службових лічильників для формування номерів

Основним зв'язком у базі даних є зв'язок між таблицею users та таблицями orders і custom_orders. Один користувач може мати багато звичайних замовлень і багато індивідуальних заявок. Для цього в таблицях orders і custom_orders використовується поле user_id, яке посилається на поле id таблиці users. Такий зв'язок дозволяє відображати в профілі користувача історію його замовлень і заявок.

Таблиця products напряду не має окремої проміжної таблиці позицій замовлення. Дані про товари, які входять до звичайного замовлення, зберігаються у полі items_json таблиці orders. Такий підхід дозволяє зберігати склад замовлення у вигляді структурованого JSON-об'єкта. У ньому можуть міститися назва товару, ціна, кількість та інші дані, необхідні для відображення замовлення. Це зручно для проєкту, оскільки замовлення зберігає інформацію про товари на момент оформлення покупки.

У таблиці custom_orders частина інформації також зберігається у JSON-полях. Поле contact_json містить контактні дані клієнта, поле params_json - параметри майбутнього виробу, а reference_urls_json - посилання на приклади або зображення, які користувач може надати для пояснення свого замовлення. Такий спосіб збереження є доцільним для індивідуальних заявок, оскільки параметри виробів можуть відрізнятися залежно від побажань клієнта.

Поля created_at і updated_at використовуються для фіксації часу створення та оновлення записів. Вони присутні в основних таблицях і дозволяють відстежувати, коли було створено або змінено товар, користувача чи замовлення. Це важливо для адміністрування системи та контролю актуальності даних.

Поле status у таблицях orders і custom_orders використовується для відображення поточного стану замовлення. Наприклад, замовлення може мати статус «Новий», «В обробці», «Виконано» або інший службовий стан. Завдяки цьому адміністратор або менеджер може відстежувати процес виконання замовлення та змінювати його стан у адміністративній частині.

Поля, пов'язані з оплатою, використовуються для збереження інформації про спосіб оплати, статус оплати, суму, дату оплати та примітки. Це дозволяє в

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						41
Зм.	Арк.	№ докум.	Підпис	Дата		

межах одного запису зберігати не лише дані про замовлення, а й інформацію про його фінансовий стан. Для розширеного обліку оплат використовуються поля `payment_entries_json`, де можуть зберігатися окремі записи про платежі.

Проектування бази даних також враховувало вимоги до безпеки. Паролі користувачів не повинні зберігатися у відкритому вигляді. Для авторизації використовується серверна частина, яка перевіряє введені дані та надає користувачу доступ відповідно до його ролі. Додатково в таблиці `users` передбачено поля для відновлення пароля: `password_reset_token_hash`, `password_reset_expires_at` і `password_reset_requested_at`.

Під час створення структури бази даних було важливо забезпечити унікальність окремих даних. Наприклад, електронна пошта та номер телефону користувача мають бути унікальними, щоб один обліковий запис можна було однозначно ідентифікувати. Номери замовлень також мають бути унікальними, що спрощує пошук і роботу адміністратора із замовленнями.

У результаті проектування було створено структуру бази даних, яка відповідає функціональним вимогам web-сайту «Лісова Майстерня». Вона забезпечує збереження користувачів, товарів, звичайних замовлень, індивідуальних заявок, статусів, оплат та службових лічильників. Така структура є достатньою для роботи інтернет-магазину та може бути розширена в майбутньому, наприклад шляхом додавання таблиць для відгуків, доставки, категорій або окремих позицій замовлення.

Отже, у межах підрозділу було розроблено структуру бази даних web-сайту «Лісова Майстерня», визначено основні таблиці, їх поля, призначення та зв'язки. Результатом етапу є ER-діаграма та детальний опис структури бази даних, що є основою для подальшої реалізації серверної частини, роботи з API та збереження інформації, необхідної для функціонування інтернет-магазину.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						42
Зм.	Арк.	№ докум.	Підпис	Дата		

2.4 Програмування сайту

Програмування web-сайту інтернет-магазину крафтових виробів «Лісова Майстерня» виконувалося як розробка повноцінної клієнт-серверної системи. Основним завданням цього етапу було об'єднати всі використані технології в єдиний програмний продукт, який забезпечує зручний інтерфейс користувача, взаємодію з базою даних, авторизацію, роботу з товарами, замовленнями, індивідуальними заявками та адміністративною частиною.

Розробка програмного продукту виконувалася у середовищі Visual Studio Code. Проєкт складається з двох основних частин: клієнтської та серверної. Клієнтська частина розміщена в папці client-vite і реалізована з використанням React, Vite, React Router, Axios та SCSS. Серверна частина розміщена в папці server-spring і створена на основі Java 17 та Spring Boot. Для роботи з базою даних використано PostgreSQL, а для взаємодії серверної частини з базою даних - Spring Data JPA.

Клієнтська частина відповідає за відображення інтерфейсу web-сайту та взаємодію користувача із системою. Вона містить сторінки головного екрана, каталогу товарів, детального перегляду товару, авторизації, реєстрації, профілю користувача, індивідуального замовлення та адміністративних розділів. Також у клієнтській частині реалізовано роботу з кошиком, формами, модальними вікнами, повідомленнями системи та переходами між сторінками.

Серверна частина відповідає за обробку HTTP-запитів, роботу з базою даних, перевірку прав доступу, створення та перевірку JWT-токенів, збереження користувачів, товарів, замовлень та індивідуальних заявок. Backend має пакетну структуру, до якої входять пакети config, domain, repository, security, service та web. Такий поділ дозволяє відокремити налаштування, сутності, репозиторії, сервіси, контролери та засоби безпеки.

Зв'язок між клієнтською та серверною частинами організовано через REST API. Клієнтська частина надсилає HTTP-запити до серверних маршрутів, а backend обробляє ці запити та повертає відповідь у форматі JSON. Для виконання

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
						43
Зм.	Арк.	№ докум.	Підпис	Дата		

запитів на frontend використовується бібліотека Axios. Основні API-запити пов'язані з отриманням товарів, реєстрацією та авторизацією користувачів, отриманням профілю, створенням замовлень, поданням індивідуальних заявок і керуванням адміністративними даними.

Під час реалізації API важливо було забезпечити однакову логіку обміну даними між frontend і backend. Клієнтська частина надсилає запити тільки до визначених серверних маршрутів, а серверна частина перевіряє отримані дані, виконує потрібну операцію та повертає відповідь клієнту. Такий підхід дозволяє відокремити інтерфейс користувача від логіки обробки даних і зробити роботу системи більш впорядкованою. Завдяки цьому сайт можна простіше підтримувати та розширювати новими функціями.

Для захисту даних і розмежування доступу в системі використовується JWT-авторизація. Після успішного входу користувача сервер формує токен, який передається на клієнтську частину. Frontend зберігає цей токен у localStorage і додає його до подальших запитів у заголовку Authorization. На серверній частині токен перевіряється спеціальним фільтром, після чого система визначає користувача та його роль.

У системі передбачено декілька ролей користувачів. Роль client призначена для звичайного покупця, який може переглядати товари, оформлювати замовлення, подавати індивідуальні заявки та працювати з профілем. Роль manager призначена для роботи із замовленнями. Роль admin має розширені права та дозволяє керувати товарами, користувачами й замовленнями.

Основна логіка програмування сайту полягає у взаємодії користувача з інтерфейсом, надсиланні запиту на сервер, обробці цього запиту backend-частиною та збереженні або отриманні даних із бази даних PostgreSQL. Наприклад, під час перегляду каталогу клієнтська частина надсилає запит до /api/products, сервер отримує товари з бази даних і повертає їх у відповідь. Під час авторизації користувач надсилає логін і пароль, сервер перевіряє ці дані та повертає JWT-токен.

Для зручної організації коду у frontend використовується компонентний

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

підхід. Кожна сторінка або частина інтерфейсу реалізована як окремий React-компонент. Це дозволяє повторно використовувати окремі елементи, спрощує підтримку коду та робить структуру клієнтської частини більш зрозумілою. Наприклад, каталог товарів, форма входу, профіль користувача, кошик і адміністративні сторінки реалізовані як окремі компоненти.

На серверній частині використовується багаторівнева структура. Контролери приймають HTTP-запити від клієнтської частини, сервіси виконують службову або бізнес-логіку, репозиторії забезпечують доступ до бази даних, а сутності описують таблиці PostgreSQL. Така структура відповідає принципам побудови Spring Boot-додатків і дозволяє розділити відповідальність між різними частинами backend.

У процесі програмування сайту також використовувалися готові фреймворки та бібліотеки. React застосовано для побудови інтерфейсу, Vite - для швидкої збірки та запуску клієнтської частини, React Router - для маршрутизації, Axios - для HTTP-запитів, Spring Boot - для створення серверної частини, Spring Security - для захисту доступу, Spring Data JPA - для роботи з базою даних, Flyway - для міграцій бази даних, а бібліотека jjwt - для створення та перевірки JWT-токенів.

Під час програмування особливу увагу було приділено узгодженій роботі клієнтської та серверної частин. Клієнтська частина відповідає за відображення інтерфейсу та надсилання запитів, а серверна - за обробку даних, перевірку користувачів, роботу з базою даних і повернення результатів. Такий поділ дозволяє зробити структуру проєкту більш зрозумілою, спростити подальше обслуговування та забезпечити можливість розширення функціоналу сайту.

Таким чином, програмування сайту «Лісова Майстерня» передбачало створення взаємопов'язаних frontend і backend частин, налаштування маршрутизації, реалізацію API-запитів, авторизації, роботи з товарами, замовленнями, індивідуальними заявками та адміністративними функціями. У результаті було створено web-додаток, який забезпечує основні функції інтернет-магазину та відповідає вимогам технічного завдання.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						45
Зм.	Арк.	№ докум.	Підпис	Дата		

2.4.1 Написання клієнтської частини

Клієнтська частина web-сайту інтернет-магазину крафтових виробів «Лісова Майстерня» реалізована з використанням React та Vite. Вона відповідає за відображення інтерфейсу користувача, переходи між сторінками, роботу з формами, взаємодію з серверною частиною через API, авторизацію користувачів, перегляд товарів, оформлення замовлень і подання індивідуальних заявок.

Основні файли клієнтської частини розміщені в папці client-vite. Для створення інтерфейсу використано компонентний підхід, за якого кожна сторінка або окрема частина інтерфейсу реалізована як окремий React-компонент. Такий підхід дозволяє зробити код більш структурованим, спростити його підтримку та повторно використовувати окремі елементи інтерфейсу на різних сторінках.

До основних сторінок клієнтської частини належать HomePage.jsx, ProductList.jsx, ProductDetail.jsx, Login.jsx, Register.jsx, ForgotPassword.jsx, ResetPassword.jsx, UserProfile.jsx та CustomOrderPage.jsx. Головна сторінка знайомить користувача з магазином, каталог товарів відображає асортимент виробів, сторінка товару містить детальну інформацію про конкретний виріб, а сторінка індивідуального замовлення дає змогу користувачу подати заявку на виготовлення виробу за власними параметрами.

Під час розробки клієнтської частини було враховано, що web-сайт повинен швидко реагувати на дії користувача без повного перезавантаження сторінок. Для цього в React-компонентах використовується стан елементів інтерфейсу, обробники подій і умовне відображення блоків. Завдяки цьому користувач може виконувати пошук товарів, відкривати форми, переглядати повідомлення системи, працювати з кошиком і переходити між сторінками у зручному режимі.

Для переходів між сторінками використовується бібліотека React Router. Маршрутизація налаштована у файлі client-vite/src/App.jsx, де визначаються основні сторінки web-сайту та умови доступу до окремих розділів.

Головна сторінка відкривається за адресою /, каталог товарів - за

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						46
Зм.	Арк.	№ докум.	Підпис	Дата		

маршрутом /products, сторінка окремого товару - за маршрутом /products/:id. Для сторінок профілю та індивідуального замовлення передбачено перевірку авторизації: якщо користувач не увійшов до системи, його буде перенаправлено на сторінку входу.

Для взаємодії клієнтської частини із серверною використовується бібліотека Axios. Щоб не створювати налаштування запитів у кожному компоненті окремо, у проєкті використано спільний API-клієнт. Він містить базову адресу backend-сервера, заголовки запитів і механізм автоматичного додавання JWT-токена до захищених запитів. Фрагмент програмного коду налаштування API-клієнта наведено в додатку В.

Однією з важливих функцій клієнтської частини є авторизація користувача. Вона реалізована у компоненті Login.jsx. Користувач вводить логін і пароль, після чого frontend надсилає запит на сервер. Якщо дані введено правильно, сервер повертає токен, роль користувача та контактні дані. Реєстрація та авторизація користувачів реалізовані у контролері UserController. Основний фрагмент коду наведено в додатку Г.

Для відображення асортименту товарів використовується компонент ProductList.jsx. Він відповідає за завантаження товарів із серверної частини, збереження отриманих даних у стані компонента та подальше відображення товарів у вигляді карток. Під час запиту можуть передаватися параметри пошуку, категорії, сортування, сторінки та кількості елементів на сторінці.

Кошик у клієнтській частині реалізований через компонент CartPanel.jsx та допоміжні функції з файлу utils/cartStorage.js. Він дозволяє користувачу додавати товари, переглядати вибрані позиції та переходити до оформлення замовлення. Оформлення замовлення виконується через компонент OrderModal.jsx, який збирає необхідні дані користувача та надсилає їх на серверний маршрут /api/orders.

Сторінка індивідуального замовлення реалізована у компоненті CustomOrderPage.jsx. Вона містить форму, за допомогою якої користувач може вказати параметри майбутнього виробу, контактні дані, побажання та посилання

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						47
Зм.	Арк.	№ докум.	Підпис	Дата		

на приклади. Після заповнення форми дані надсилаються на серверний маршрут `/api/custom-orders`, де зберігаються як індивідуальна заявка.

Сторінка профілю користувача реалізована у компоненті `UserProfile.jsx`. Вона призначена для відображення персональної інформації, ролі користувача, кількості замовлень і відомостей про звичайні та індивідуальні заявки. Для отримання даних профілю клієнтська частина звертається до серверного маршруту `/api/users/profile`.

Під час програмування клієнтської частини також було реалізовано роботу зі службовими повідомленнями. Вони використовуються для інформування користувача про успішне виконання дій або про помилки. Наприклад, повідомлення можуть з'являтися під час неправильного введення даних авторизації, невдалого запиту до сервера, успішного створення замовлення або надсилання індивідуальної заявки.

Важливою частиною клієнтської логіки є перевірка доступу до сторінок. Неавторизований користувач може переглядати головну сторінку, каталог і сторінку товару, однак доступ до профілю та індивідуального замовлення вимагає входу в систему. Для цього в маршрутах використовується перевірка наявності ролі користувача. Якщо користувач не авторизований, він перенаправляється на сторінку входу.

Таким чином, клієнтська частина web-сайту «Лісова Майстерня» забезпечує зручну взаємодію користувача з інтернет-магазином. Вона реалізує маршрутизацію, авторизацію, перегляд каталогу, сторінку товару, кошик, оформлення замовлення, профіль користувача та індивідуальні заявки. Використання React, Vite, React Router, Axios і SCSS дозволило створити сучасний, гнучкий і зручний інтерфейс, який взаємодіє із серверною частиною через REST API.

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
						48
Зм.	Арк.	№ докум.	Підпис	Дата		

2.4.2 Написання admin-частини

Адміністративна частина web-сайту «Лісова Майстерня» призначена для керування основними даними інтернет-магазину. Вона забезпечує роботу з товарами, користувачами, звичайними замовленнями та індивідуальними заявками. На відміну від клієнтської частини, яка орієнтована на перегляд товарів і оформлення замовлень, admin-частина має службове призначення та використовується адміністратором або менеджером для підтримки роботи сайту.

У проєкті адміністративна частина реалізована як частина клієнтського React-додатку, але доступ до її сторінок обмежується залежно від ролі користувача. Для адміністратора доступні сторінки керування товарами, користувачами та замовленнями, а для менеджера - робота із замовленнями. Такий поділ дозволяє розмежувати повноваження користувачів і запобігти несанкціонованому доступу до службових функцій.

До основних компонентів адміністративної частини належать AdminProductList.jsx, AdminUserPage.jsx та AdminOrdersUnifiedPage.jsx. Компонент AdminProductList.jsx відповідає за відображення списку товарів, пошук, сортування, створення, редагування, видалення та перегляд статистики товарів. Компонент AdminUserPage.jsx використовується для перегляду користувачів, створення нового користувача, редагування ролі, активації або деактивації облікового запису. Компонент AdminOrdersUnifiedPage.jsx призначений для об'єднаного керування звичайними та індивідуальними замовленнями.

Доступ до адміністративних сторінок обмежується на рівні маршрутизації. У клієнтській частині використовується перевірка ролі користувача, завдяки чому звичайний клієнт не може перейти до сторінок адміністрування. Наприклад, маршрути для керування товарами та користувачами доступні тільки адміністратору, а сторінка замовлень може бути доступна як адміністратору, так і менеджеру.

На серверній частині перевірка прав доступу виконується додатково. Це

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

важливо, оскільки обмеження лише на frontend не є достатнім для захисту системи. Backend перевіряє користувача та його роль перед виконанням службових операцій. Наприклад, перед створенням нового товару система перевіряє, чи авторизований користувач і чи має він роль адміністратора.

Захист адміністративної частини забезпечується перевіркою JWT-токена та ролі користувача. Для цього використовується серверний фільтр AuthFilter, який зчитує заголовок Authorization, перевіряє токен і передає дані користувача для подальшої перевірки доступу. Фрагмент програмного коду перевірки JWT-токена наведено в додатку Ж.

Керування товарами реалізовано у контролері ProductController, який забезпечує створення, редагування та видалення товарів. Фрагмент коду наведено в додатку Д. Метод відповідає за створення нового товару в адміністративній частині. Перед збереженням товару виконується перевірка користувача за допомогою WebSupport.requireUser(request) та перевірка ролі адміністратора через WebSupport.requireAdmin(user). Після цього формується об'єкт ProductEntity, йому присвоюється ідентифікатор, а запис зберігається в базі даних за допомогою productRepository.save(p).

Сторінка керування товарами є однією з основних сторінок admin-частини. Вона дозволяє адміністратору додавати нові товари, редагувати вже створені позиції, видаляти товари, переглядати список виробів, виконувати пошук і сортування. Для роботи з товарами використовуються API-запити GET /api/products, GET /api/products/admin/stats, POST /api/products, PUT /api/products/{id} та DELETE /api/products/{id}. Завдяки цьому адміністратор може підтримувати каталог у актуальному стані без внесення змін до програмного коду.

Для відображення товарів у базі даних на серверній частині використовується JPA-сутність ProductEntity. Вона описує структуру таблиці products і містить основні поля товару: ідентифікатор, назву, матеріали, розміри, ціну та статус активності.

Сторінка керування користувачами реалізована у компоненті AdminUserPage.jsx. Вона дозволяє адміністратору переглядати список

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		50

zareestrovanih korystuvachiv, stvorjувати novі oblikovі zapisi, zmінjувати rolь korystuvacha, aktivувати або deaktivувати oblikovі zapisi та peregljадати пов'язану інформацію. Для цього використовуються API-запити GET /api/users, GET /api/users/stats, GET /api/users/{id}, POST /api/users і PUT /api/users/{id}.

Kерування користувачами є важливою частиною адміністративного функціоналу, оскільки саме через ролі визначається рівень доступу до системи. Звичайний користувач має доступ до клієнтської частини, менеджер може працювати із замовленнями, а адміністратор має повні права керування магазином. Такий поділ дозволяє уникнути ситуацій, коли звичайний користувач отримує доступ до службових операцій.

Окреме місце в admin-частині займає сторінка керування замовленнями AdminOrdersUnifiedPage.jsx. Вона об'єднує роботу зі звичайними замовленнями та індивідуальними заявками. Це зручно для адміністратора або менеджера, оскільки всі звернення клієнтів можна переглядати в одному розділі. У цій частині використовуються запити GET /api/orders/unified-feed, GET /api/orders/unified-kpis, PUT /api/orders/{id}, PATCH /api/orders/{id}/status, PUT /api/custom-orders/{id} і PATCH /api/custom-orders/{id}/status.

Робота із замовленнями та індивідуальними заявками виконується через серверні контролери OrderController і CustomOrderController. Вони відповідають за перегляд замовлень, зміну статусів, додавання службових коментарів, створення індивідуальної заявки та збереження її параметрів у базі даних. Фрагменти програмного коду роботи із замовленнями та заявками наведено в додатку Е.

Для зручної роботи із замовленнями в системі використовується генерація порядкових номерів. Це дозволяє присвоювати кожному замовленню зрозумілий номер, який можна використовувати під час пошуку, обробки та комунікації з клієнтом. Генерація номера виконується через сервіс OrderNumberService.

Метод next(), який відповідає за отримання наступного номера замовлення. Для цього використовується службова таблиця counters. Якщо лічильник ще не створений, формується новий запис із початковим значенням. Далі поле seq

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
						51
Зм.	Арк.	№ докум.	Підпис	Дата		

збільшується на одиницю, після чого оновлений запис зберігається в базі даних. Анотація @Transactional забезпечує виконання операції як єдиної транзакції.

Важливою частиною admin-функціоналу є перевірка авторизації. Оскільки адміністративні запити змінюють дані в базі, доступ до них повинен бути захищений. Для цього на клієнтській частині до кожного запиту додається JWT-токен, а на серверній частині він перевіряється у фільтрі AuthFilter.

Для прикладу візьмемо фільтр авторизації. Спочатку з HTTP-запиту зчитується заголовок Authorization. Якщо заголовок відсутній або не містить Bearer-токен, запит передається далі без авторизаційних даних. Якщо токен наявний, він перевіряється через JwtService, після чого з бази даних отримується користувач. Надалі ці дані можуть використовуватися для перевірки ролі та доступу до службових функцій.

Admin-частина тісно пов'язана із серверними контролерами та репозиторіями. Контролери приймають HTTP-запити від клієнтської частини, перевіряють права доступу, обробляють отримані дані та звертаються до репозиторіїв. Репозиторії, у свою чергу, виконують операції з базою даних, наприклад пошук, збереження, оновлення або видалення записів. Такий підхід дозволяє розділити логіку інтерфейсу, обробки запитів і роботи з базою даних.

Завдяки реалізації адміністративної частини адміністратор може повноцінно керувати роботою інтернет-магазину. Він має можливість оновлювати каталог товарів, змінювати інформацію про користувачів, переглядати замовлення, змінювати їх статуси та працювати з індивідуальними заявками. Менеджер, у свою чергу, може працювати із замовленнями без доступу до керування товарами й користувачами.

Таким чином, admin-частина web-сайту «Лісова Майстерня» забезпечує службові функції, необхідні для підтримки роботи інтернет-магазину. Вона реалізує керування товарами, користувачами, звичайними замовленнями та індивідуальними заявками, а також використовує механізми JWT-авторизації та перевірки ролей для захисту адміністративних операцій.

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
						52
Зм.	Арк.	№ докум.	Підпис	Дата		

2.5 Тестування web-сайту

Тестування web-сайту інтернет-магазину крафтових виробів «Лісова Майстерня» є завершальним етапом розробки технічного та робочого проєкту. Його основною метою є перевірка правильності роботи основних функцій програмного продукту, відповідності реалізованого функціоналу технічному завданню, коректності взаємодії клієнтської та серверної частин, а також виявлення можливих помилок у роботі сайту.

У процесі тестування перевірялися як клієнтська частина, реалізована за допомогою React та Vite, так і серверна частина, створена на основі Spring Boot. Окрему увагу було приділено роботі з базою даних PostgreSQL, авторизації користувачів за допомогою JWT, розмежуванню доступу між ролями, роботі каталогу товарів, оформленню звичайних замовлень, поданню індивідуальних заявок та функціонуванню адміністративної частини.

Тестування виконувалося вручну шляхом послідовної перевірки сторінок сайту та основних сценаріїв використання. Для перевірки клієнтської частини використовувався браузер, у якому відкривалися сторінки web-сайту та виконувалися дії користувача. Для перевірки серверної частини використовувалися запити до API, а також перевірка результатів збереження даних у базі даних. У процесі тестування аналізувалися повідомлення системи, коректність переходів між сторінками, правильність обробки введених даних і поведінка сайту в разі помилкових дій користувача.

Під час тестування були використані такі види перевірки: функціональне тестування, тестування інтерфейсу користувача, тестування форм, тестування авторизації та ролей доступу, інтеграційне тестування frontend і backend, тестування роботи з базою даних, а також перевірка адміністративної частини. Такий підхід дозволив комплексно оцінити роботу сайту та переконатися, що основні функції виконуються правильно.

Функціональне тестування проводилося для перевірки основних можливостей сайту. Було протестовано відкриття головної сторінки, перегляд

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

каталогу товарів, відкриття сторінки детального перегляду товару, додавання товарів до кошика, оформлення замовлення, створення індивідуальної заявки, реєстрація, авторизація, перегляд профілю користувача та робота адміністративної панелі.

Тестування інтерфейсу користувача передбачало перевірку зручності використання сайту. Було перевірено правильність розміщення кнопок, форм, таблиць, карток товарів, повідомлень системи та навігаційних елементів. Також перевірялося, чи зрозуміло користувачу, які дії він може виконати на кожній сторінці, та чи не виникає труднощів під час переходу між розділами.

Окремо виконувалося тестування форм. Перевірялися форми авторизації, реєстрації, відновлення пароля, оформлення замовлення та подання індивідуальної заявки. У процесі перевірки тестувалися як правильні, так і помилкові вхідні дані. Наприклад, перевірялося, як система реагує на незаповнені обов'язкові поля, неправильний пароль, відсутність контактних даних або некоректно введену інформацію.

Важливим етапом було тестування авторизації та розмежування доступу. Було перевірено, що неавторизований користувач може переглядати публічні сторінки, але не має доступу до профілю, індивідуального замовлення та адміністративної частини. Після авторизації користувач отримує доступ до функцій відповідно до своєї ролі. Звичайний користувач може працювати з клієнтською частиною, менеджер - із замовленнями, а адміністратор - із товарами, користувачами та замовленнями.

Інтеграційне тестування проводилося для перевірки взаємодії клієнтської та серверної частин. Під час такої перевірки frontend надсилав HTTP-запити до backend, а серверна частина обробляла їх і повертала відповідь у форматі JSON. Перевірялося отримання товарів із каталогу, надсилання даних авторизації, створення замовлення, подання індивідуальної заявки, отримання профілю користувача та робота адміністративних запитів.

Для чіткої організації процесу перевірки було сформовано план тестування, у якому визначено основні об'єкти, мету та спосіб виконання

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		54

перевірки. Узагальнений план тестування web-сайту наведено в таблиці 2.3.

Таблиця 2.3 - План тестування web-сайту «Лісова Майстерня»

Об'єкт тестування	Мета перевірки	Спосіб перевірки
Клієнтські сторінки сайту	Перевірити відкриття головної сторінки, каталогу, сторінки товару та профілю	Послідовно відкрити сторінки в браузері та перевірити відображення елементів
Форми користувача	Перевірити авторизацію, реєстрацію, відновлення пароля та введення даних	Заповнити форми коректними й помилковими даними
Каталог і замовлення	Перевірити перегляд товарів, роботу кошика та оформлення покупки	Додати товар до кошика, заповнити форму та створити тестове замовлення
Індивідуальні заявки	Перевірити подання заявки на виготовлення виробу за параметрами клієнта	Заповнити форму індивідуального замовлення та перевірити її надсилання
Адміністративна частина	Перевірити керування товарами, користувачами й замовленнями	Увійти під роллю адміністратора та виконати основні дії з даними
Ролі та доступ	Перевірити розмежування доступу між користувачем, менеджером і адміністратором	Спробувати відкрити захищені сторінки під різними ролями

План тестування дозволяє охопити основні частини web-сайту: клієнтський інтерфейс, форми, каталог, замовлення, індивідуальні заявки та адміністративну панель. Такий підхід дає змогу перевірити не лише окремі сторінки, а й повні сценарії роботи користувача із системою.

Крім плану тестування, під час перевірки було складено контрольний список основних компонентів web-сайту. Основні критерії перевірки подано в таблиці 2.4.

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

Таблиця 2.4 - Контрольний список перевірки компонентів web-сайту

№	Компонент або функція	Критерії перевірки
1	Навігація сайту	Коректні переходи між основними сторінками та розділами
2	Каталог товарів	Завантаження товарів, відображення карток, пошук і сортування
3	Сторінка товару	Відображення назви, опису, ціни, зображення та характеристик
4	Авторизація та реєстрація	Вхід до системи, створення облікового запису, реакція на помилки
5	Кошик і замовлення	Додавання товару, оформлення покупки та передача даних на сервер
6	Індивідуальна заявка	Заповнення параметрів виробу та надсилання заявки
7	Адміністративна частина	Керування товарами, користувачами, замовленнями та статусами
8	Розмежування доступу	Заборона доступу до службових сторінок без відповідної ролі

Контрольний список показує, що під час тестування було перевірено основні функції web-сайту, які впливають на роботу користувача, адміністратора та менеджера. Детальніші результати перевірки описано текстом після таблиці, щоб не перевантажувати її зайвими рядками.

Під час тестування головної сторінки було перевірено її відкриття, відображення інформаційних блоків, кнопок переходу та навігаційних елементів. Сторінка відкривалася без помилок, а всі основні елементи були доступні користувачу. Це підтверджує коректність верстки та правильну роботу маршрутизації.

Під час тестування каталогу товарів перевірялося завантаження даних із серверної частини. Товари відображалися у вигляді карток, користувач міг переглянути основну інформацію про виріб і перейти до детальної сторінки. Також перевірялася робота сторінки товару, де відображаються зображення, опис, ціна та характеристики виробу.

Тестування авторизації показало, що користувач може увійти в систему за

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
						56
Зм.	Арк.	№ докум.	Підпис	Дата		

правильними даними. У разі введення неправильного пароля або незаповнення обов'язкових полів система виводить повідомлення про помилку. Після успішного входу користувач отримує доступ до сторінок, які потребують авторизації.

Під час перевірки реєстрації було протестовано створення нового облікового запису. Після введення необхідних даних система передавала їх на серверну частину, де створювався новий користувач. Також було перевірено, що система реагує на некоректні або вже використані дані.

Оформлення замовлення перевірялося шляхом додавання товару до кошика та подальшого підтвердження покупки. Після оформлення замовлення дані передавалися на backend і зберігалися в базі даних. У результаті адміністратор або менеджер міг переглянути створене замовлення в адміністративній частині.

Окремо було перевірено подання індивідуальної заявки. Користувач заповнював форму, зазначав контактні дані, параметри майбутнього виробу та додаткові побажання. Після надсилання заявки дані зберігалися на сервері й ставали доступними для обробки в адміністративній частині.

Тестування адміністративної частини проводилося для перевірки роботи сторінок керування товарами, користувачами та замовленнями. Було перевірено, що адміністратор може додавати нові товари, редагувати наявні записи, переглядати користувачів і змінювати статуси замовлень. Менеджер, у свою чергу, має доступ до роботи із замовленнями, але не має повних прав адміністратора.

Також було виконано перевірку безпеки доступу. Неавторизований користувач не повинен мати можливості перейти до профілю, індивідуального замовлення або адміністративних сторінок. Звичайний користувач не повинен отримувати доступ до сторінок керування товарами, користувачами та замовленнями. Така перевірка підтверджує правильність роботи ролей і JWT-авторизації.

Окремо під час тестування перевірялася обробка помилкових дій

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

користувача. До таких ситуацій належить введення неправильних даних авторизації, спроба надіслати незаповнену форму, відкриття захищеної сторінки без входу в систему або виконання дії без відповідних прав доступу. У таких випадках система повинна не завершувати роботу з помилкою, а виводити зрозуміле повідомлення та не дозволяти виконати некоректну операцію.

Також було перевірено стабільність взаємодії між клієнтською та серверною частинами. Під час виконання основних дій контролювалося, чи правильно frontend надсилає запити до backend, чи повертає сервер очікувану відповідь і чи зберігаються дані в базі даних. Така перевірка є важливою, оскільки web-сайт працює як єдина система, де помилка в одному модулі може вплинути на роботу інших функцій.

Для підтвердження результатів тестування копії екранів роботи окремих сторінок web-сайту доцільно подати в додатках. До таких зображень можуть належати головна сторінка, каталог товарів, сторінка авторизації, профіль користувача, форма індивідуального замовлення та адміністративна сторінка керування замовленнями.

За результатами тестування встановлено, що основні функції web-сайту працюють коректно. Користувач може переглядати товари, авторизуватися, оформлювати замовлення, подавати індивідуальні заявки та переглядати власний профіль. Адміністратор може керувати товарами, користувачами та замовленнями, а менеджер - працювати із замовленнями. Виявлені під час тестування незначні недоліки можуть бути усунені в процесі подальшого вдосконалення програмного продукту.

Отже, тестування web-сайту «Лісова Майстерня» підтвердило працездатність основних функцій програмного продукту та його відповідність поставленим вимогам. Проведені перевірки показали, що сайт може використовуватися як повноцінний інтернет-магазин крафтових виробів із клієнтською, менеджерською та адміністративною частинами.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

3 СПЕЦІАЛЬНИЙ РОЗДІЛ

3.1 Інструкція з розміщення сайту в Інтернеті

Розміщення web-сайту в мережі Інтернет є важливим етапом підготовки програмного продукту до практичного використання. Після завершення розробки та тестування інтернет-магазин крафтових виробів «Лісова Майстерня» потрібно розмістити на хостингу, налаштувати серверну частину, підключити базу даних та перевірити працездатність сайту за публічною адресою.

Розроблений web-сайт має клієнт-серверну архітектуру. Клієнтська частина створена за допомогою React та Vite і розміщена окремо від серверної частини. Серверна частина реалізована на Spring Boot і відповідає за обробку запитів, авторизацію користувачів, роботу з товарами, замовленнями, індивідуальними заявками та базою даних PostgreSQL. Такий поділ дозволяє розгортати frontend і backend на різних платформах, що спрощує підтримку та масштабування проєкту.

Для розміщення клієнтської частини було використано платформу Vercel[13]. Вона призначена для швидкого розгортання frontend-застосунків, зокрема проєктів, створених за допомогою React та Vite. Клієнтська частина web-сайту «Лісова Майстерня» розміщена за адресою - <https://forest-workshop1-perfectqqs-projects.vercel.app/>

Серверна частина web-сайту розміщена на платформі Render[14] як Spring Boot-застосунок. На цій самій платформі створено PostgreSQL-базу даних, яка використовується для збереження користувачів, товарів, замовлень та індивідуальних заявок. Точну адресу backend-сервісу та внутрішні параметри підключення до бази даних у звіті не наведено з міркувань безпеки.

Загальний принцип розміщення сайту полягає в тому, що користувач відкриває клієнтську частину через браузер за адресою frontend, розміщеного на платформі Vercel. Після цього frontend надсилає HTTP-запити до backend-сервісу на платформі Render, а серверна частина, своєю чергою, виконує операції з

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

PostgreSQL-базою даних. Взаємодію між користувачем, клієнтською частиною, серверною частиною та базою даних показано на рисунку 3.1.

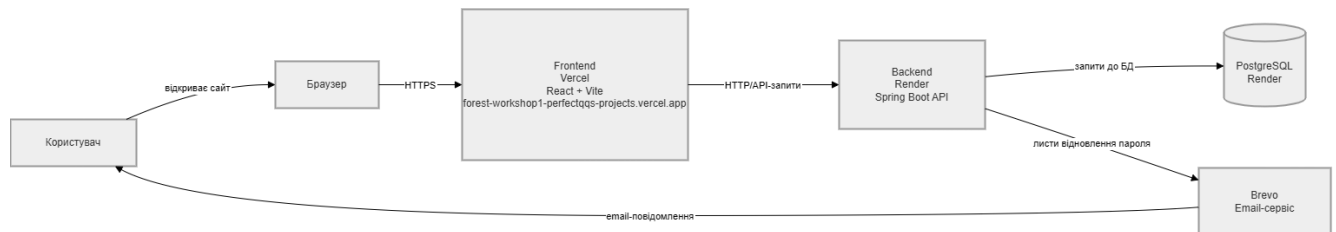


Рисунок 3.1 — Структура розміщення web-сайту в Інтернеті

Для розгортання web-сайту використовується кілька взаємопов'язаних компонентів: frontend, backend, база даних, поштовий сервіс та змінна для підключення клієнтської частини до API. Їх призначення узагальнено в таблиці 3.1.

Таблиця 3.1 - Компоненти розміщення web-сайту «Лісова Майстерня»

Елемент системи	Платформа	Призначення
Frontend	Vercel	Розміщення клієнтської частини React/Vite
Backend	Render	Запуск серверної частини Spring Boot
База даних	Render PostgreSQL	Збереження користувачів, товарів, замовлень і заявок
Email-сервіс	Brevo	Надсилення листів для відновлення пароля
API-з'єднання	VITE_API_URL	Підключення frontend до backend

Frontend відповідає за відображення інтерфейсу користувача, backend реалізує бізнес-логіку та обробку запитів, база даних забезпечує збереження інформації, а сервіс Brevo використовується для надсилення електронних листів. API-з'єднання між клієнтською та серверною частинами налаштовується через змінну середовища VITE_API_URL.

Перед розміщенням frontend-частини необхідно виконати збірку проєкту. Для цього спочатку потрібно перейти до папки клієнтської частини командою cd

client-vite, після чого встановити залежності командою `npm install` і створити production-збірку командою `npm run build`.

Після виконання команди `npm run build` формується папка `dist`, яка містить готові статичні файли для розгортання на платформі Vercel. Для коректної роботи маршрутизації React Router використовується конфігурація `vercel.json`, яка забезпечує перенаправлення всіх маршрутів на файл `index.html`.

Для підключення frontend до серверної частини використовується змінна середовища `VITE_API_URL`, яка містить адресу backend-сервісу на платформі Render. Завдяки цьому клієнтська частина може надсилати запити до API після розміщення в Інтернеті. Якщо ця змінна не задана, frontend за замовчуванням намагатиметься звертатися до локальної адреси `http://localhost:8081`, що підходить лише для локального запуску.

Серверна частина проєкту збирається за допомогою Maven. Для створення готового JAR-файлу Spring Boot-застосунку використовується команда `mvn -f server-spring/pom.xml clean package`. Після завершення збірки застосунк може бути запущений як JAR-файл за допомогою команди `java -jar server-spring/target/server-spring-0.0.1-SNAPSHOT.jar`.

Крім стандартного запуску, проєкт підтримує контейнеризацію за допомогою Docker. Для цього використовується файл `Dockerfile`, який автоматизує процес збірки та запуску серверної частини. Такий спосіб є зручним для хмарних платформ, оскільки дозволяє запускати backend у підготовленому контейнерному середовищі.

Для коректної роботи сайту після розміщення необхідно виконати базове налаштування середовища виконання та параметрів взаємодії між його компонентами. Зокрема, налаштовуються параметри підключення до бази даних, механізми автентифікації та авторизації користувачів, а також мережеві параметри, що забезпечують обмін даними між клієнтською та серверною частинами застосунку.

Важливою складовою процесу розгортання є забезпечення безпечного зберігання конфігураційних параметрів. Для цього сучасні хмарні платформи

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

надають можливість використовувати змінні середовища, які дозволяють відокремити службові налаштування від програмного коду та уникнути розміщення конфіденційної інформації у відкритому доступі.

Оскільки клієнтська та серверна частини системи розміщені на різних платформах, додатково виконується налаштування політики міждоменного доступу (CORS). Це забезпечує можливість безпечного обміну даними між frontend і backend через HTTP-запити без порушення обмежень, що накладаються браузером.

Також під час розгортання налаштовуються параметри, необхідні для роботи функцій автентифікації користувачів, відновлення пароля та інших сервісних можливостей системи. Завдяки цьому забезпечується стабільна робота web-сайту в мережі Інтернет та коректна взаємодія всіх його компонентів.

База даних PostgreSQL розміщується на платформі Render. Після створення бази даних платформа надає параметри підключення, які використовуються серверною частиною через змінні середовища. Під час запуску застосунку автоматично виконуються Flyway-міграції, що створюють необхідну структуру таблиць для користувачів, товарів, замовлень, індивідуальних заявок і службових лічильників.

Під час розміщення web-сайту важливо було забезпечити доступність усіх частин системи після перенесення з локального середовища в Інтернет. Для цього клієнтська частина повинна коректно звертатися до віддаленого backend-сервісу, а серверна частина — мати доступ до PostgreSQL-бази даних. Окремо перевірялося, чи правильно працюють змінні середовища, які містять адреси сервісів і параметри підключення. Це дозволяє уникнути ситуацій, коли сайт відкривається, але не може отримати товари, виконати авторизацію або зберегти замовлення.

Після завершення налаштування виконується розгортання frontend на платформі Vercel та backend на платформі Render. Загальна послідовність розміщення web-сайту складається з підготовки проекту, створення бази даних, налаштування backend, розгортання frontend і перевірки взаємодії між усіма частинами системи.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						62
Зм.	Арк.	№ докум.	Підпис	Дата		

Покроково процес розміщення web-сайту «Лісова Майстерня» можна подати так:

1. Підготувати проєкт до розгортання.
2. Створити PostgreSQL-базу даних на платформі Render.
3. Розгорнути серверну частину та налаштувати необхідні параметри.
4. Розгорнути клієнтську частину на платформі Vercel.
5. Налаштувати змінну VITE_API_URL.
6. Перевірити взаємодію між frontend і backend.
7. Виконати тестування основних функцій web-сайту.

Окрему увагу під час розміщення web-сайту потрібно приділити перевірці змінних середовища. Неправильно вказана адреса backend-сервісу або бази даних може призвести до того, що сайт відкриватиметься, але не зможе завантажувати товари, виконувати авторизацію або створювати замовлення. Тому після налаштування Vercel і Render необхідно перевірити, чи правильно вказано VITE_API_URL, параметри підключення до PostgreSQL, JWT-секрет і дозволені джерела запитів через CORS_ALLOWED_ORIGINS.

Також важливо врахувати, що frontend і backend розгортаються окремо, тому помилка в одній частині системи може впливати на роботу всього web-сайту. Наприклад, якщо серверна частина тимчасово недоступна, користувач зможе відкрити головну сторінку, але каталог товарів, профіль, авторизація або оформлення замовлення працюватимуть некоректно. Саме тому після розгортання потрібно перевіряти не лише відкриття сторінок, а й повні сценарії взаємодії користувача із сайтом.

Після успішного розміщення доцільно періодично перевіряти працездатність хостингу, доступність backend-сервісу та стан бази даних. У разі оновлення програмного коду необхідно повторно виконати збірку клієнтської частини, оновити серверну частину та переконатися, що після змін не порушилася робота основних функцій. Такий підхід дозволяє підтримувати стабільну роботу web-сайту після його публікації в Інтернеті.

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
						63
Зм.	Арк.	№ докум.	Підпис	Дата		

3.2 Інструкція з обслуговування та наповнення сайту

Після розміщення web-сайту в мережі Інтернет необхідно забезпечити його подальше обслуговування та регулярне наповнення. Для інтернет-магазину крафтових виробів «Лісова Майстерня» це передбачає підтримку актуального каталогу товарів, обробку замовлень, перегляд індивідуальних заявок, керування користувачами та контроль роботи адміністративної частини.

Обслуговування сайту виконується через адміністративну частину. Вхід до системи здійснюється через сторінку авторизації /login, де користувач вводить email або номер телефону та пароль. Після успішного входу система визначає роль користувача і відкриває доступ до відповідних розділів. Звичайний користувач із роллю client працює з клієнтською частиною сайту, менеджер із роллю manager має доступ до замовлень, а адміністратор із роллю admin може керувати товарами, користувачами та замовленнями.

Основне меню адміністративної частини складається з розділів керування товарами, користувачами та замовленнями. Розділ товарів використовується для наповнення каталогу, розділ користувачів - для перегляду облікових записів і зміни ролей, а розділ замовлень - для обробки звичайних замовлень та індивідуальних заявок. Менеджер працює переважно із замовленнями, тоді як адміністратор має доступ до всіх службових розділів.

Для додавання нового товару адміністратор відкриває сторінку керування товарами, створює новий запис і заповнює основні поля: назву виробу, ціну, матеріали, розміри, категорію, опис, фотографію, кількість на складі та статус активності. Після збереження дані передаються на серверну частину та записуються до бази даних. Під час наповнення каталогу важливо використовувати зрозумілі назви, якісні зображення та повні описи виробів, оскільки для крафтової продукції зовнішній вигляд і характеристики мають велике значення.

Редагування існуючого товару виконується в тому самому розділі. Адміністратор може змінити назву, опис, ціну, категорію, кількість на складі,

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
						64
Зм.	Арк.	№ докум.	Підпис	Дата		

фотографію або статус активності. Якщо товар тимчасово недоступний, його доцільно не видаляти, а деактивувати. У такому разі запис залишається в базі даних, але не відображається звичайним користувачам у каталозі.

Керування користувачами здійснюється через окрему сторінку адміністративної частини. У цьому розділі адміністратор може переглядати список зареєстрованих користувачів, відкривати дані облікового запису, змінювати роль, активувати або деактивувати користувача. Зміна ролі повинна виконуватися обережно, оскільки від неї залежить доступ до службових функцій сайту.

Окремим напрямом обслуговування є робота із замовленнями. У системі передбачено об'єднану сторінку, на якій адміністратор або менеджер може переглядати звичайні замовлення з каталогу та індивідуальні заявки. Для кожного замовлення доступні контактні дані клієнта, склад замовлення, сума, статус, інформація про оплату та службові коментарі. Під час обробки замовлення адміністратор або менеджер може змінювати його статус і додавати примітки.

Індивідуальні заявки використовуються для замовлення виробів за параметрами клієнта. У такій заявці можуть бути зазначені розміри, матеріал, побажання щодо вигляду, контактні дані та посилання на приклади. Після перегляду заявки менеджер або адміністратор уточнює можливість виконання, за потреби додає коментар і змінює статус обробки.

Під час експлуатації сайту адміністратор повинен регулярно перевіряти актуальність каталогу, статуси замовлень і правильність даних користувачів. Нові товари потрібно додавати з повним описом і фотографіями, недоступні позиції - деактивувати, а завершені замовлення - переводити у відповідний статус. Такий порядок обслуговування дозволяє підтримувати сайт в актуальному стані.

Для стабільної роботи web-сайту важливо врахувати мінімальні технічні вимоги до середовища розміщення. Оскільки frontend розміщено на Vercel, а backend і база даних - на Render, значна частина інфраструктури обслуговується хмарними платформами. Основні вимоги до середовища роботи сайту наведено в таблиці 3.2.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						65
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.2 - Мінімальна конфігурація середовища для роботи web-сайту

Параметр	Мінімальна вимога	Призначення
Операційна система	Linux-середовище або хмарна платформа	Запуск backend-застосунку
Процесор	1 vCPU	Обробка HTTP-запитів
Оперативна пам'ять	від 1 GB, бажано 2 GB	Стабільна робота Spring Boot
Дисковий простір	від 10 GB	Збереження службових файлів і логів
Java	17	Запуск серверної частини
Node.js[15]	сумісна версія для Vite	Запуск клієнтської частини
PostgreSQL	хмарна або окрема БД	Збереження даних сайту
Порти	8081, 443	Робота backend і HTTPS-доступ

Крім технічної конфігурації, потрібно підтримувати правильні значення змінних середовища: параметри підключення до PostgreSQL, JWT-секрет, адреси frontend і backend, а також налаштування CORS, не зберігаючи конфіденційні дані у відкритому коді або звіті.

Важливою частиною обслуговування є резервне копіювання бази даних. Оскільки на момент підготовки звіту окремий механізм регулярного резервного копіювання не налаштований, у подальшому доцільно використовувати автоматичні резервні копії PostgreSQL у Render або періодичний експорт бази даних.

Також потрібно періодично перевіряти працездатність frontend, backend і бази даних, а в разі помилок перевірити доступність backend-сервісу, правильність VITE_API_URL, налаштування CORS і стан бази даних.

Отже, обслуговування та наповнення web-сайту «Лісова Майстерня» передбачає підтримку каталогу товарів, користувачів, замовлень та індивідуальних заявок. Адміністративна частина забезпечує зручне керування вмістом сайту, а правильна конфігурація середовища - його стабільну роботу в мережі Інтернет.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						66
Зм.	Арк.	№ докум.	Підпис	Дата		

3.3 Інструкція з популяризації та підтримки сайту

Після розміщення web-сайту в мережі Інтернет важливо не лише забезпечити його технічну працездатність, а й організувати подальшу популяризацію. Для інтернет-магазину крафтових виробів «Лісова Майстерня» просування має важливе значення, оскільки сайт орієнтований на продаж виробів ручної роботи, залучення нових клієнтів і приймання індивідуальних замовлень.

Основною метою популяризації web-сайту є збільшення кількості відвідувачів, підвищення впізнаваності майстерні, залучення потенційних покупців і формування довіри до магазину. Оскільки крафтові вироби часто обирають за зовнішнім виглядом, якістю матеріалів і унікальністю, просування сайту повинно поєднувати технічну оптимізацію, якісне наповнення каталогу, активність у соціальних мережах і регулярне оновлення інформації.

Одним із головних напрямів популяризації є пошукова оптимізація сайту. Вона передбачає правильне оформлення назв сторінок, описів товарів, заголовків, текстового наповнення та ключових слів. Для магазину «Лісова Майстерня» доцільно використовувати запити, пов'язані з дерев'яними виробами, крафтовою продукцією, handmade-товарами, декором для дому, меблями ручної роботи та індивідуальними замовленнями.

Під час наповнення каталогу кожен товар повинен мати зрозумілу назву, детальний опис, ціну, категорію, характеристики та якісне зображення. У назвах товарів бажано використовувати конкретні формулювання, наприклад: «дерев'яна полиця ручної роботи», «декор з дерева», «крафтовий виріб для дому», «подарунок з дерева». Такий підхід допомагає користувачам швидше знаходити потрібний товар і покращує індексацію сторінок пошуковими системами.

Описи товарів мають бути не лише короткими, а й інформативними. У них доцільно зазначати матеріал, розміри, призначення виробу, особливості виготовлення, можливість індивідуального замовлення та рекомендації з використання. Для крафтових виробів це особливо важливо, оскільки покупець хоче розуміти, з чого виготовлений товар, як він виглядає в інтер'єрі та чи можна

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		67

змінити його параметри.

Важливу роль у популяризації сайту відіграють фотографії. Для кожного товару бажано додавати якісні зображення з різних ракурсів, а також фото виробу в реальному використанні. Якщо товар виготовляється вручну, доцільно показувати не лише готовий результат, а й частину процесу створення. Це підвищує довіру покупця та підкреслює унікальність продукції.

Окремим напрямом просування є використання соціальних мереж. Для майстерні, що виготовляє крафтові вироби, найбільш доречними є Instagram, Facebook і TikTok. У соціальних мережах можна публікувати фотографії готових виробів, короткі відео процесу виготовлення, приклади індивідуальних замовлень, відгуки клієнтів і посилання на сторінки товарів у каталозі. Такі публікації допомагають залучати користувачів на сайт і формувати постійну аудиторію.

Ще одним способом популяризації є контент-маркетинг. На сайті або в соціальних мережах можна публікувати корисні матеріали, пов'язані з дерев'яними виробами. Наприклад, це можуть бути поради щодо догляду за виробами з дерева, ідеї декору для дому, приклади оформлення інтер'єру, переваги handmade-продукції або рекомендації щодо вибору подарунків ручної роботи.

Крім просування, важливо забезпечити регулярну підтримку сайту. Адміністратор повинен стежити за актуальністю товарів, вчасно оновлювати ціни, видаляти або деактивувати недоступні позиції, додавати нові вироби й перевіряти правильність описів. Каталог товарів має бути актуальним, оскільки застаріла інформація може знизити довіру користувача до магазину.

Підтримка сайту також передбачає контроль роботи замовлень. Адміністратор або менеджер повинен регулярно переглядати нові замовлення, змінювати їх статуси, відповідати на індивідуальні заявки та додавати коментарі у разі потреби. Швидка реакція на звернення клієнтів підвищує якість обслуговування та збільшує ймовірність повторних замовлень.

Для покращення взаємодії з клієнтами доцільно збирати відгуки після виконання замовлень. Відгуки можна використовувати в соціальних мережах або

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						68
Зм.	Арк.	№ докум.	Підпис	Дата		

на сайті для підтвердження якості роботи майстерні. Також вони допомагають зрозуміти, які товари користуються найбільшим попитом і які елементи обслуговування потрібно покращити.

Технічна підтримка web-сайту включає перевірку доступності frontend, backend і бази даних. Якщо сайт відкривається, але не завантажує каталог або не виконує авторизацію, потрібно перевірити роботу backend-сервісу, змінну VITE_API_URL, налаштування CORS і стан PostgreSQL-бази даних. Також необхідно періодично переглядати журнали помилок на хостингу.

Окрему увагу потрібно приділяти безпеці. Паролі, токени, ключі доступу та параметри підключення до бази даних не повинні зберігатися у відкритому коді або публічному репозиторії. Для цього використовуються змінні середовища на платформах Vercel і Render. Також потрібно контролювати, щоб доступ до адміністративної частини мали лише користувачі з відповідними ролями.

Важливою частиною підтримки є резервне копіювання бази даних. Оскільки база даних містить інформацію про користувачів, товари, замовлення та індивідуальні заявки, її втрата може негативно вплинути на роботу магазину. Тому доцільно налаштувати автоматичні резервні копії PostgreSQL або виконувати періодичний експорт даних.

Після внесення змін до функціоналу або дизайну сайту потрібно перевіряти основні сценарії роботи: відкриття головної сторінки, перегляд каталогу, авторизацію, оформлення замовлення, подання індивідуальної заявки та роботу адміністративної частини. Це дозволяє переконатися, що оновлення не порушило працездатність web-сайту.

Отже, популяризація та підтримка web-сайту «Лісова Майстерня» повинні виконуватися комплексно. Для залучення користувачів потрібно використовувати SEO-оптимізацію, якісне наповнення каталогу, соціальні мережі та контент-маркетинг. Для стабільної роботи сайту необхідно регулярно оновлювати інформацію, контролювати замовлення, перевіряти технічний стан системи та забезпечувати резервне копіювання даних.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		69

4 ЕКОНОМІЧНИЙ РОЗДІЛ

Метою економічної частини даного дипломного проєкту є проведення економічних розрахунків, спрямованих на визначення економічної ефективності розробки інтернет-магазину крафтових виробів «Лісова Майстерня», прийняття рішення про подальший розвиток і впровадження або ж недоцільність проведення відповідної розробки.

Об'єктом розробки є інтернет-магазин крафтових виробів «Лісова Майстерня»

Розрахунок вартості розробки виконується в декілька етапів:

- описати технологічний процес розробки із зазначенням трудомісткості кожної операції;
- визначити суму витрат на оплату праці основного і допоміжного персоналу, включаючи відрахування на соціальні заходи;
- обчислити витрати на електроенергію;
- нарахувати суму амортизаційних відрахувань;
- визначити суму накладних витрат;
- скласти кошторис та визначити собівартість робіт;
- розрахувати ціну робіт;
- визначити економічну ефективність та термін окупності.

Економічні розрахунки дають змогу оцінити не лише витрати на створення програмного продукту, а й доцільність його подальшого використання. Для інтернет-магазину «Лісова Майстерня» важливо визначити, скільки ресурсів потрібно для розробки, які статті витрат є основними та за який період проєкт може окупитися після впровадження.

Оскільки web-сайт має клієнтську, серверну та адміністративну частини, витрати на його створення пов'язані переважно з оплатою праці розробників, використанням комп'ютерної техніки, електроенергії та організаційними витратами. Отримані показники використовуються для визначення собівартості, ціни розробки та економічної ефективності проєкту.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						70
Зм.	Арк.	№ докум.	Підпис	Дата		

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення НДР

У цьому підрозділі розглянуто основні етапи технологічного процесу розробки інтернет-магазину «Лісова Майстерня». Проєкт передбачає створення каталогу товарів, бази даних, авторизації користувачів, кошика, оформлення замовлень, адміністративного керування та перевірку роботи API.

Для визначення загальної тривалості робіт витрати часу за окремими операціями технологічного процесу зведено у таблицю 4.1.

Таблиця 4.1 – Середній час виконання робіт та стадії технологічного процесу розробки інтернет-магазину «Лісова Майстерня»

№ п/п	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
1	Планування та аналіз	Кер. проєкту Рm	9
		Інженер (l1)	8
2	Розробка технічного завдання	Кер. проєкту (Рm)	7
		Інженер (l1)	6
3	Дизайн інтерфейсу	Інженер (l1)	14
		Інженер (l2)	24
4	Розробка функціоналу	Інженер (l1)	40
5	Тестування та відладка	Тестувальник	10
6	Документування	Інженер (l1)	3
7	Розгортання та підтримка	Інженер (l2)	15
8	Управління проєктом	Кер. проєкту (Рm)	24
Разом			160

Сумарний час виконання операцій технологічного процесу становить 160 годин. Такий обсяг часу пояснюється тим, що розробка інтернет-магазину «Лісова Майстерня» охоплює не лише створення зовнішнього вигляду сторінок, а й реалізацію повноцінної клієнт-серверної взаємодії, бази даних, авторизації користувачів та адміністративної частини.

Найбільша частина часу припадає на програмування клієнтської та серверної частин, оскільки саме на цих етапах реалізується основна логіка роботи інтернет-магазину. До неї належать перегляд каталогу, оформлення замовлень, робота з індивідуальними заявками, авторизація користувачів, обробка запитів до бази даних та адміністрування товарів, користувачів і замовлень.

Окреме значення має етап проєктування структури сайту та бази даних. Перед початком програмування необхідно визначити основні сторінки, ролі користувачів, логіку переходів, склад таблиць бази даних і зв'язки між ними. Це дозволяє уникнути значних змін під час реалізації та забезпечити послідовну розробку всіх частин програмного продукту.

Час, відведений на верстку сторінок, використовується для створення зручного інтерфейсу користувача. На цьому етапі формуються головна сторінка, каталог товарів, сторінка товару, форми авторизації, профіль користувача та адміністративні сторінки. Важливо, щоб усі сторінки мали єдиний стиль оформлення та були зручними для використання.

Окремий час виділено на тестування, оскільки перевірка працездатності сайту є важливою умовою приймання готового програмного продукту. Тестування дозволяє виявити помилки у формах, маршрутизації, роботі з базою даних, авторизації та адміністративних функціях до моменту впровадження сайту.

Отже, визначена тривалість технологічного процесу є обґрунтованою, оскільки враховує всі основні етапи створення web-сайту: аналіз, проєктування, верстку, програмування, тестування та підготовку програмного продукту до використання.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						72
Зм.	Арк.	№ докум.	Підпис	Дата		

4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

У даному підрозділі проводиться розрахунок витрат, пов'язаних з оплатою праці та відрахуваннями на соціальні заходи, що необхідні для розробки інтернет-магазину крафтових виробів «Лісова Майстерня». Для розрахунку прийнято склад виконавців, який відповідає характеру вебпроєкту: керівник проєкту, full-stack розробник, front-end розробник і тестувальник.

Основна заробітна плата розраховується за формулою:

$$Z_{\text{осн.}} = T_c \cdot K_r \quad (4.1)$$

де: T_c – тарифна ставка, грн. (приймаємо для керівника проєкту (Рм) – 450 грн./год, інженера (І2) – 272 грн./год.), інженера (І1) – 113 грн./год., тестувальник – 100 грн./год.; K_r – кількість відпрацьованих годин.

Отже, основна заробітна плата для:

Керівника проєкту (Рм) $Z_{\text{осн.}} = 40 \cdot 450 = 18\,000$ грн.

Інженера (І2) $Z_{\text{осн.}} = 39 \cdot 272 = 10\,608$ грн.

Інженера (І1) $Z_{\text{осн.}} = 71 \cdot 113 = 8\,023$ грн.

Тестувальник $Z_{\text{осн.}} = 10 \cdot 100 = 1\,000$ грн.

Сумарна основна заробітна плата становить

$$Z_{\text{осн.}} = 18\,000 + 10\,608 + 8\,023 + 1\,000 = 37\,631 \text{ грн.}$$

Додаткова заробітна плата становить 10 – 15 % від суми основної заробітної плати.

$$Z_{\text{дод.}} = Z_{\text{осн.}} \cdot K_{\text{допл.}} \quad (4.2)$$

де: $K_{\text{допл.}}$ – коефіцієнт додаткових виплат працівникам.

Отже додаткова заробітна плата по категоріях працівників становить:

Керівника проєкту $Z_{\text{дод.}} = 18\,000 \cdot 0,1 = 1\,800$ грн.

Інженера (І2) $Z_{\text{дод.}} = 10\,608 \cdot 0,1 = 1\,061$ грн.

Інженера (І1) $Z_{\text{дод.}} = 8\,023 \cdot 0,1 = 802$ грн.

Тестувальник $Z_{\text{дод.}} = 1\,000 \cdot 0,1 = 100$ грн.

Загальна додаткова заробітна плата становить:

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		73

$$З_{\text{дод}} = 1\,800 + 1\,061 + 802 + 100 = 3\,763 \text{ грн.}$$

Звідси загальні витрати на оплату праці ($B_{\text{о.п.}}$) визначаються за формулою:

$$B_{\text{о.п.}} = З_{\text{осн.}} + З_{\text{дод.}} \quad (4.3)$$

$$B_{\text{о.п.}} = 37\,631 + 3\,763 = 41\,394 \text{ грн.}$$

Єдиний соціальний внесок (ЄСВ – 22%) визначається за формулою:

$$B_{\text{ЄСВ}} = B_{\text{оп}} * 0,22 \quad (4.4)$$

$$B_{\text{ЄСВ}} = 41\,394 * 0,22 = 9\,107 \text{ грн.}$$

Проведені розрахунки витрат на оплату праці наведено у таблиці 4.2.

Таблиця 4.2 – Зведені розрахунки витрат на оплату праці

№ п / п	Категорія працівників	Основна заробітна плата, грн.			Додатков а заробітна плата, грн.	ЄСВ, грн.	Всього витрати на оплату праці, грн. $6 = 3+4+5$
		Тарифн а ставка, грн.	К-сть годин	Фактично нарах. зарплати, грн.			
		1	2	3	4	5	6
1	Кер. проєкту (Рm)	450	40	18 000	1 800		
2	Інженера (І2)	272	39	10 608	1 061		
3	Інженера (І1)	113	71	8 023	802		
4	Тестувальни к	100	10	1 000	100		
Разом				37 631	3 763	9 107	50 501

Отже, загальні витрати на оплату праці становлять 50 501 грн.

Отримані витрати на оплату праці є найбільшою складовою загальної вартості розробки. Це пояснюється тим, що створення web-сайту потребує виконання аналітичних, проєктних, програмних і тестових робіт. У розрахунку враховано участь фахівців, які виконують різні завдання.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						74
Зм.	Арк.	№ докум.	Підпис	Дата		

4.3 Розрахунок витрат на електроенергію

Розрахуємо вартість електроенергії. Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$З_{\text{в}} = W * T * S \quad (4.5)$$

де: W – необхідна потужність, кВт; T – кількість годин роботи обладнання;
 S – вартість кіловат-години електроенергії (приймаємо 15, 94 грн).

У нашій системі використовується 1 ПК, який застосовується для виконання основних етапів розробки інтернет-магазину «Лісова Майстерня». На ньому здійснюється написання програмного коду, запуск клієнтської та серверної частин, робота з базою даних, тестування функціоналу, підготовка документації та перевірка працездатності web-сайту. Витрати на електроенергію для цього комп'ютера обчислимо окремо, взявши за основу, що час роботи обладнання залежить від тривалості виконуваних робіт згідно з таблицею 4.1, а споживана потужність комп'ютера становить 0,82 кВт/год.

$$З_{\text{ек}} = 0,82 * 160 * 15,94 = 2\,091 \text{ грн.}$$

Витрати на електроенергію становлять 2 091 грн.

Витрати на електроенергію порівняно з витратами на оплату праці є значно меншими, однак вони також враховуються у загальній собівартості. Під час розробки комп'ютер використовується для написання програмного коду, роботи з базою даних, запуску локального сервера, тестування сайту та підготовки документації.

Урахування витрат на електроенергію дозволяє зробити кошторис повнішим, оскільки процес розробки програмного забезпечення потребує постійного використання технічних засобів. Тому ця стаття витрат включається до загальної суми витрат на створення інтернет-магазину.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						75
Зм.	Арк.	№ докум.	Підпис	Дата		

4.4 Розрахунок суми амортизаційних відрахувань для інтернет-магазину крафтових виробів «Лісова Майстерня»

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення

Комп'ютери та оргтехніка належать до четвертої групи основних фондів.

Амортизація на них нараховується лише в випадку, якщо мінімально допустимі строки їх корисного використання 2 роки. Для визначення амортизаційних відрахувань застосовуємо формулу:

$$A = \frac{B_B * H_A}{100\%} * T, \quad (4.6)$$

де: А – амортизаційні відрахування за звітний період, грн.;

Б_В – балансова вартість групи основних фондів на початок звітного періоду, грн.;

Н_А – норма амортизації, 0,04 %.

Оскільки для написання програми та її тестування використовується один ПК, вартістю 50000,00 грн., то сума амортизаційних відрахувань становитиме:

$$A = \frac{50\,000,00 * 0,04}{150} * 160 = 2\,133 \text{ грн.}$$

Амортизаційні відрахування враховують поступове зношення комп'ютерної техніки, що використовується під час розробки. Хоча програмний продукт створюється переважно інтелектуальною працею, для його реалізації необхідне технічне обладнання, зокрема персональний комп'ютер.

Урахування амортизації дає змогу точніше визначити собівартість розробки, оскільки частина вартості обладнання переноситься на виконані роботи та входить до загальних витрат проєкту.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						76
Зм.	Арк.	№ докум.	Підпис	Дата		

4.5 Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 20–60 % від суми основної та додаткової заробітної плати працівників.

$$H_B = V_{o.p.} \cdot 0,2..0,6 \quad (4.7)$$

де: H_B – накладні витрати.

$$H_B = 41\,394 \cdot 0,4 = 16\,558 \text{ грн.}$$

Накладні витрати охоплюють витрати, які прямо не пов'язані з написанням програмного коду, але необхідні для організації процесу розробки. До них можна віднести витрати на обслуговування робочого місця, використання інтернет-з'єднання, організаційне забезпечення, супровід документації та інші непрямі витрати.

Для даного проекту накладні витрати прийнято у відсотковому співвідношенні до витрат на оплату праці. Такий підхід дозволяє врахувати допоміжні витрати без деталізації кожної дрібної статті окремо та отримати більш реалістичну загальну вартість розробки.

У процесі розробки інтернет-магазину «Лісова Майстерня» накладні витрати також пов'язані з підготовкою робочого середовища, налаштуванням програмних засобів, перевіркою працездатності застосунку та збереженням проміжних результатів роботи. Такі витрати не можна безпосередньо віднести до окремого модуля сайту, однак вони забезпечують нормальне виконання всього технологічного процесу.

Урахування накладних витрат є необхідним для повнішого визначення собівартості програмного продукту. Якщо враховувати лише оплату праці, електроенергію та амортизацію, розрахунок буде неповним, оскільки частина організаційних і допоміжних витрат залишиться поза кошторисом. Тому накладні витрати включаються до загальної суми витрат на розробку інтернет-магазину.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						77
Зм.	Арк.	№ докум.	Підпис	Дата		

4.6 Складання кошторису витрат та визначення собівартості інтернет-магазину крафтових виробів «Лісова Майстерня»

Для складання кошторису витрат та визначення собівартості, результати проведених вище розрахунків зведемо у таблиці 4.4.

Таблиця 4.4 - Кошторис витрат на розробку інтернет-магазину крафтових виробів «Лісова Майстерня»

№	Зміст витрат	Сума, грн.	В % до загальної суми
1.	Витрати на оплату праці	50 501	71
2.	Витрати на електроенергію	2 091	3
3.	Амортизаційні відрахування	2 133	3
4.	Накладні витрати	16 558	23
5.	Собівартість	71 283	100

Собівартість (C_B) НДР розрахуємо за формулою:

$$C_B = B_{o.p} + B_{c.z} + Z_e + A + H_b \quad (4.8)$$

Отже, собівартість дорівнює $C_B = 71\,283$ грн.

Собівартість розробки показує мінімальну суму витрат, необхідну для створення інтернет-магазину «Лісова Майстерня» без урахування прибутку та податкових нарахувань. До неї входять витрати на оплату праці, соціальні відрахування, електроенергію, амортизацію та накладні витрати.

Отримане значення собівартості використовується як основа для подальшого розрахунку ціни розробки. Саме від правильності визначення собівартості залежить обґрунтованість кінцевої вартості програмного продукту та оцінка його економічної доцільності.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						78
Зм.	Арк.	№ докум.	Підпис	Дата		

4.7 Розрахунок ціни розробки інтернет-магазину крафтових виробів «Лісова Майстерня»

Розрахунок ціни науково-дослідної роботи включає в себе урахування різноманітних факторів, таких як рівень рентабельності, собівартість та податкова ставка.

Ціну робіт можна визначити за формулою:

$$Ц = C_{\text{в}} * (1 + P_{\text{рен}}) * (1 + \text{ПДВ}), \quad (4.9)$$

де: $C_{\text{в}}$ – собівартість; $P_{\text{рен}}$ – рівень рентабельності; ПДВ – ставка податку на додану вартість.

$$Ц = 71\,283 * (1 + 0,3) * (1 + 0,2) = 111\,201 \text{ грн.}$$

Розрахунок ціни розробки дає змогу визначити орієнтовну вартість створення інтернет-магазину «Лісова Майстерня» як готового програмного продукту. На відміну від собівартості, ціна враховує не лише понесені витрати, а й запланований прибуток, що робить розробку економічно доцільною для виконавця.

У вартість розробки входить комплекс робіт, пов'язаних зі створенням клієнтської частини, серверної логіки, бази даних, адміністративної панелі, системи авторизації, тестування та підготовки сайту до розміщення в мережі Інтернет. Тому кінцева ціна відображає не окремий елемент сайту, а повний цикл створення web-застосунку.

Під час визначення ціни також враховується те, що розроблений сайт може використовуватися не лише як навчальний проєкт, а й як основа для реального інтернет-магазину. У разі розвитку проєкту додаткові витрати будуть пов'язані переважно з розширенням функціоналу, а не з повною повторною розробкою сайту.

Отримане значення ціни може бути використане як орієнтир у разі замовної розробки подібного інтернет-магазину. Воно показує, яку суму необхідно передбачити для створення програмного продукту з аналогічною структурою, функціоналом і технічними можливостями.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						79
Зм.	Арк.	№ докум.	Підпис	Дата		

4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Для визначення ефективності продукту розраховують чисту теперішню вартість (ЧТВ) і термін окупності (Ток).

$$\text{ЧТВ} = -\text{Св} + \sum_{i=1}^t \frac{\Gamma_{\text{п}}}{(1+i)^t}, \quad (4.10)$$

де: Св – собівартість розробки; $\Gamma_{\text{п}}$ – грошовий потік за t – ий рік; t – відповідний рік проекту; i – величина дисконтної ставки (10...15%).

$$\text{ЧТВ} = -71\,283 + \frac{39\,918}{(1+0,1)^1} + \frac{39\,918}{(1+0,1)^2} + \frac{39\,918}{(1+0,1)^3} = 27\,738 \text{ грн}$$

Якщо $\text{ЧТВ} \geq 0$, то проект може бути рекомендований до впровадження.

Термін окупності визначається за формулою:

$$T_{\text{ок}} = T_{\text{пв}} + \frac{H_{\text{в}}}{\Gamma_{\text{пр}}} \quad (4.11)$$

де: $T_{\text{пв}}$ – період до повного відшкодування витрат, років; $H_{\text{в}}$ – невідшкодовані витрати на початок року, грн.; $\Gamma_{\text{пр}}$ – грошовий потік на початок року, грн.

$$T_{\text{ок}} = 2 + \frac{2\,275}{39\,918} = 2,1 \text{ р.}$$

Основні результати економічних розрахунків узагальнено у таблиці техніко-економічних показників. Її наведено у графічній частині кваліфікаційної роботи на 4 аркуші 2026.КВР.122.423.03.00.00 ТБ. Прибутковість проекту та термін окупності свідчать про його фінансову ефективність і здатність повернути вкладені кошти приблизно протягом 2,1 року. Отже, на основі отриманих показників можна зробити висновок, що розробка інтернет-магазину крафтових виробів «Лісова Майстерня» є доцільною з економічної точки зору.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						80
Зм.	Арк.	№ докум.	Підпис	Дата		

5 ОХОРОНА ПРАЦІ

5.1 Основні заходи по запобіганню травматизму та професійних захворювань

Під час виконання дипломного проєкту «Розробка інтернет-магазину крафтових виробів “Лісова Майстерня”» основна робота пов’язана з проєктуванням, програмуванням, тестуванням і документуванням вебдодатку. Робочий процес передбачає тривале використання персонального комп’ютера або ноутбука, роботу з монітором, клавіатурою, мишею, мережевим обладнанням, базою даних і середовищем розробки. Така діяльність не належить до фізично важкої праці, однак має власні небезпечні та шкідливі фактори.

Охорона праці у межах даного проєкту розглядається як система організаційних, технічних, санітарно-гігієнічних і профілактичних заходів. Її метою є недопущення травмування працівника під час користування електрообладнанням, запобігання професійним захворюванням, зменшення зорового й нервово-емоційного навантаження та підтримання безпечних умов для стабільної роботи над програмним продуктом.

Робоче місце розробника інтернет-магазину «Лісова Майстерня» належить до робочих місць операторського типу, оскільки більшість операцій виконується сидячи та потребує постійної концентрації уваги. Працівник працює з програмним кодом, структурою бази даних, адміністративною частиною сайту, тестовими формами, документацією та сервісами перевірки API. Тому важливо правильно організувати робочий простір, забезпечити достатнє освітлення, підтримувати належний мікроклімат і дотримуватися режиму праці та відпочинку.

До основних факторів ризику належать несправність електричного обладнання, пошкодження кабелів, перевантаження подовжувачів, захаращення робочої зони проводами, неправильна посадка, недостатнє освітлення, тривале статичне навантаження, зорове напруження, психоемоційна втома та незадовільні параметри мікроклімату. Їх тривалий вплив підвищує ризик перевтоми,

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
						81
Зм.	Арк.	№ докум.	Підпис	Дата		

погіршення самопочуття та помилкових дій.

Професійні захворювання у працівників, які тривалий час працюють за комп'ютером, найчастіше пов'язані із зоровим перенапруженням, порушеннями постави, болем у спині та шиї, напруженням м'язів кистей, головним болем, втомою нервової системи та зниженням працездатності, тому для розробника особливо важливим є попередження перевтоми органів зору під час роботи з кодом, таблицями, формами введення даних і тестовими сценаріями.

Основні небезпечні та шкідливі фактори, характерні для робочого місця розробника, а також можливі наслідки їх впливу наведено в таблиці 5.1.

Таблиця 5.1 – Небезпечні та шкідливі фактори на робочому місці

Фактор ризику	Можливий наслідок	Профілактичний захід
Несправні розетки, кабелі або блоки живлення	Ураження електричним струмом, коротке замикання, займання	Використання справного обладнання, перевірка кабелів, мережевий фільтр
Захаращення робочого місця проводами	Спотикання, падіння, пошкодження техніки	Упорядкування кабелів, звільнення проходів, фіксація проводів
Тривала робота за монітором	Перевтома очей, сухість очей, головний біль	Перерви, вправи для очей, правильна яскравість і відстань до екрана
Неправильна посадка	Біль у спині, шиї, плечах, порушення постави	Ергономічний стілець, правильна висота столу, підтримка спини
Нервово-емоційне навантаження	Перевтома, зниження концентрації, помилки в роботі	Планування роботи, чергування завдань, короткі перерви

Організаційні заходи щодо запобігання травматизму передбачають перевірку робочого місця перед початком роботи, підтримання порядку на столі, правильне розміщення техніки та дотримання правил безпечної експлуатації електрообладнання. Робоча поверхня повинна бути чистою, стійкою та достатньою для розміщення ноутбука або монітора, клавіатури, миші й робочих матеріалів.

Особливу увагу потрібно приділяти електробезпеці. Комп'ютерна техніка, маршрутизатори, зарядні пристрої та периферійне обладнання працюють від електромережі, тому забороняється використовувати пошкоджені кабелі, розетки зі слідами перегріву, несправні подовжувачі та блоки живлення невідомого походження. У разі появи запаху горілого, іскріння або надмірного нагрівання роботу необхідно припинити та від'єднати пристрій від мережі.

Ергономіка робочого місця є важливою умовою профілактики професійних захворювань. Монітор слід розміщувати на відстані приблизно 50–70 см від очей, а верхня межа екрана має бути на рівні очей або трохи нижче. Клавіатура і миша повинні знаходитися на зручній висоті, щоб руки перебували у природному положенні. Якщо робота виконується за ноутбуком, доцільно використовувати підставку, окрему клавіатуру та мишу.

Стілець повинен забезпечувати підтримку попереку, стійке положення тіла та можливість регулювання висоти. Ноги мають стояти на підлозі або підставці, а спина повинна мати опору. Тривале сидіння без зміни положення тіла спричиняє статичне навантаження на м'язи, тому протягом робочого дня необхідно вставати, розминати шию, плечі, спину та кисті рук.

Для попередження зорового перенапруження необхідно контролювати освітлення та параметри екрана. Світло має бути достатнім, але не створювати відблисків на моніторі. Яскравість і контрастність екрана потрібно налаштовувати відповідно до освітлення приміщення. Під час перерв доцільно переводити погляд на віддалені предмети, часто моргати та виконувати прості вправи для очей.

Раціональна організація роботи також зменшує нервово-емоційне навантаження. Розробку інтернет-магазину доцільно поділяти на окремі етапи:

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						83
Зм.	Арк.	№ докум.	Підпис	Дата		

створення структури бази даних, реалізація серверної логіки, налаштування маршрутизації, тестування авторизації, перевірка кошика та оформлення замовлення. Такий підхід дозволяє уникати хаотичної роботи та своєчасно виявляти помилки.

Комплекс основних заходів безпечної організації роботи наведено в таблиці 5.2.

Таблиця 5.2 - Комплекс заходів безпечної організації роботи

Група заходів	Зміст заходів	Очікуваний результат
Організаційні	Планування робочого часу, дотримання перерв, підтримання порядку	Зменшення втоми та кількості помилок
Технічні	Використання справної техніки, перевірка кабелів, недопущення перегріву	Запобігання електротравмам і пошкодженню обладнання
Ергономічні	Правильне розміщення монітора, клавіатури, миші та стільця	Профілактика болю у спині, шії та руках
Санітарно-гігієнічні	Провітрювання, контроль мікроклімату, прибирання пилу, належне освітлення	Комфортні умови праці та підтримання працездатності

Для запобігання травматизму забороняється розміщувати рідини поруч із ноутбуком, системним блоком, клавіатурою, мережевим обладнанням і розетками. Потрапляння води або напоїв на електронні пристрої може спричинити коротке замикання, пошкодження обладнання або ураження електричним струмом. Також необхідно регулярно прибирати робоче місце,

оскільки пил погіршує санітарний стан приміщення та може негативно впливати на охолодження комп'ютерної техніки.

Таким чином, основними заходами запобігання травматизму та професійним захворюванням під час розробки інтернет-магазину «Лісова Майстерня» є дотримання правил електробезпеки, правильна організація робочого місця, забезпечення ергономіки, контроль освітлення й мікроклімату, регулярні перерви, підтримання чистоти та раціональне планування роботи.

Під час роботи над програмним продуктом важливо правильно організувати робоче місце. Стіл повинен мати достатню площу для комп'ютера, клавіатури, миші та матеріалів. Монітор слід встановлювати так, щоб верхня частина екрана була приблизно на рівні очей, а відстань до нього становила 50–70 см. Це зменшує навантаження на зір і сприяє правильному положенню тіла.

Важливим заходом профілактики професійних захворювань є дотримання режиму праці та відпочинку. Тривала робота за комп'ютером може спричинити втому очей, напруження м'язів спини, шиї та рук. Тому необхідно робити короткі перерви, змінювати положення тіла, виконувати вправи для очей і розминку для рук.

Також необхідно дотримуватися правил електробезпеки. Комп'ютерна техніка повинна бути справною, а кабелі не мають створювати небезпеку спотикання. Забороняється ремонтувати обладнання під напругою, торкатися оголених проводів або використовувати несправні розетки. У разі несправності роботу потрібно припинити та повідомити відповідальну особу.

Окрему увагу слід приділяти пожежній безпеці. У приміщенні не можна перевантажувати електромережу, використовувати пошкоджені подовжувачі або залишати ввімкнену техніку без нагляду. Робоче місце повинно утримуватися в порядку, а легкозаймисті матеріали не мають розміщуватися біля джерел живлення та нагрівання.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		85

5.2 Визначення параметрів мікроклімату

Мікроклімат робочого приміщення є важливим чинником, що впливає на самопочуття, працездатність і безпеку працівника. Для робочого місця, на якому виконується розробка та тестування інтернет-магазину «Лісова Майстерня», мікроклімат має особливе значення, оскільки програмування потребує тривалої концентрації уваги, стабільної роботи комп'ютерної техніки та комфортних умов перебування у приміщенні.

Під мікрокліматом розуміють сукупність параметрів повітряного середовища у робочій зоні. До основних параметрів належать температура повітря, відносна вологість, швидкість руху повітря, температура поверхонь і наявність теплового випромінювання. Для офісного робочого місця розробника найбільш важливими є температура, вологість і швидкість руху повітря.

Робота над дипломним проєктом належить до легких робіт категорії Іа, оскільки виконується переважно сидячи та не потребує значного фізичного напруження. Водночас вона пов'язана з нервово-емоційним навантаженням, аналізом інформації, пошуком помилок у програмному коді та тестуванням функцій. Нормативні значення параметрів мікроклімату для легкої роботи категорії Іа наведено в таблиці 5.3.

Таблиця 5.3 - Оптимальні параметри мікроклімату для легкої роботи категорії Іа

Період року	Температура повітря, °С	Відносна вологість, %	Швидкість руху повітря, м/с
Холодний період року	22–24	40–60	не більше 0,1
Теплий період року	23–25	40–60	не більше 0,1

Оцінювання мікроклімату виконується шляхом вимірювання параметрів

повітря у робочій зоні та порівняння їх із нормативними значеннями. Температуру й вологість визначають термогігрометром, швидкість руху повітря - анемометром, а показники можуть змінюватися через опалення, провітрювання та роботу техніки.

Для розрахунку прийнято умовне приміщення, у якому виконується розробка дипломного проєкту. У ньому розміщено робоче місце з комп'ютерною технікою, а джерелами тепловиділення є обладнання, освітлення та людина. Результати вимірювання наведено в таблиці 5.4.

Таблиця 5.4 - Результати вимірювання параметрів мікроклімату

Час вимірювання	Температура, °C	Відносна вологість, %	Швидкість руху повітря, м/с	Примітка
09:00	22,1	50	0,07	Початок роботи, приміщення провітрене
13:00	23,0	47	0,08	Середина робочого дня, техніка працює стабільно
17:00	23,4	45	0,09	Кінець робочого дня, температура підвищилась
Середнє значення	22,8	47,3	0,08	Показники перебувають у межах норми

Середнє значення температури визначається як середнє арифметичне трьох вимірювань: $t_{\text{сер}} = (22,1 + 23,0 + 23,4) / 3 = 22,8 \text{ }^{\circ}\text{C}$. Отримане значення входить до оптимального діапазону для холодного періоду року, який становить 22–24 °С для легкої роботи категорії Іа. Це означає, що температурний режим робочого місця є комфортним і не створює додаткового теплового навантаження.

Середня відносна вологість повітря становить $\varphi_{\text{сер}} = (50 + 47 + 45) / 3 = 47,3 \text{ } \%$, а середня швидкість руху повітря - $v_{\text{сер}} = (0,07 + 0,08 + 0,09) / 3 = 0,08 \text{ м/с}$. Обидва показники відповідають оптимальним умовам: вологість перебуває в діапазоні 40–60 %, а швидкість руху повітря не перевищує 0,1 м/с.

Порівняння фактичних показників із нормативними значеннями наведено в таблиці 5.5.

Таблиця 5.5 - Оцінка відповідності параметрів мікроклімату нормативним значенням

Параметр	Нормативне значення	Фактичне середнє значення	Висновок
Температура повітря	22–24 °С	22,8 °С	Відповідає оптимальним умовам
Відносна вологість	40–60 %	47,3 %	Відповідає оптимальним умовам
Швидкість руху повітря	не більше 0,1 м/с	0,08 м/с	Відповідає оптимальним умовам

За результатами порівняння можна зробити висновок, що параметри мікроклімату робочого приміщення відповідають оптимальним умовам для виконання легкої розумової роботи. Температура, вологість і швидкість руху повітря перебувають у межах рекомендованих значень, тому не створюють

додаткового ризику для здоров'я працівника та не погіршують умови виконання дипломного проєкту.

Для підтримання стабільних умов необхідно регулярно провітрювати приміщення, контролювати роботу опалення, уникати перегріву робочої зони прямими сонячними променями, очищувати вентиляційні отвори комп'ютерної техніки та підтримувати чистоту. У теплий період року доцільно використовувати жалюзі, кондиціонер або вентилятор так, щоб потік повітря не був спрямований безпосередньо на працівника.

Крім основних параметрів мікроклімату, важливо враховувати загальний стан робочої зони. Робоче місце не повинно бути розміщене біля джерел протягів, надмірного тепла або прямих сонячних променів. Нерівномірний розподіл температури в приміщенні може створювати дискомфорт, що негативно впливає на концентрацію уваги під час роботи з комп'ютером.

Також необхідно стежити за тим, щоб у приміщенні не накопичувалося надмірне тепло від техніки. Комп'ютер, монітор, маршрутизатор та інші пристрої під час роботи виділяють тепло, тому робочу зону потрібно періодично провітрювати. Це дозволяє підтримувати нормальний повітрообмін і створює сприятливі умови для тривалої розумової праці.

Таким чином, визначення параметрів мікроклімату є важливою складовою оцінки умов праці під час виконання дипломного проєкту. Для робочого місця розробника інтернет-магазину «Лісова Майстерня» основними контрольованими параметрами є температура повітря, відносна вологість і швидкість руху повітря. Дотримання оптимальних параметрів мікроклімату разом із заходами профілактики травматизму створює безпечні та комфортні умови для якісного виконання роботи.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						89
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено web-сайт інтернет-магазину крафтових виробів «Лісова Майстерня». Програмний продукт призначений для продажу дерев'яних виробів ручної роботи, оформлення замовлень, подання індивідуальних заявок, роботи з профілем користувача та адміністрування даних магазину.

У першому розділі було проведено аналітичний огляд існуючих рішень у сфері електронної комерції. Було розглянуто інтернет-магазини, маркетплейси, соціальні мережі, конструктори сайтів і CRM-системи. На основі аналізу визначено доцільність створення власного web-сайту для майстерні, оскільки він дозволяє самостійно керувати каталогом, замовленнями, користувачами та індивідуальними заявками. Також було сформовано технічне завдання, визначено функціональні вимоги, етапи розробки та порядок тестування програмного продукту.

У другому розділі було виконано розробку технічного та робочого проєкту. Було спроектовано структуру web-сайту, визначено основні сторінки, ролі користувачів, логіку переходів і загальний шаблон сторінок. Також описано створення та верстку головної сторінки, каталогу, сторінки товару, сторінки індивідуального замовлення, профілю користувача та адміністративної частини.

Для збереження даних було розроблено структуру бази даних PostgreSQL. У базі даних передбачено таблиці для користувачів, товарів, звичайних замовлень, індивідуальних заявок і службових лічильників. Структуру бази даних було подано за допомогою ER-діаграми.

Під час програмування клієнтську частину реалізовано з використанням React, Vite, React Router, Axios та SCSS, а серверну частину - на основі Spring Boot, Spring Security, Spring Data JPA та Flyway. Реалізовано каталог товарів, кошик, профіль користувача, індивідуальні заявки, API-запити, JWT-авторизацію та взаємодію з базою даних. Адміністративна частина забезпечує керування товарами, користувачами, звичайними замовленнями та індивідуальними

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						90
Зм.	Арк.	№ докум.	Підпис	Дата		

заявками, а поділ ролей дозволяє обмежити доступ до службових функцій сайту.

Також у другому розділі було проведено тестування web-сайту. Перевірено роботу основних сторінок, форм, каталогу товарів, авторизації, реєстрації, кошика, оформлення замовлення, індивідуальних заявок, профілю користувача та адміністративної частини. За результатами тестування встановлено, що основні функції сайту працюють коректно.

У третьому розділі було розглянуто розміщення, обслуговування та підтримку web-сайту. Клієнтську частину розміщено на платформі Vercel, а серверну частину та базу даних PostgreSQL - на платформі Render. Було описано налаштування взаємодії між frontend і backend, змінні середовища, CORS, порядок перевірки роботи сайту після розгортання, а також інструкцію з обслуговування адміністративної частини, наповнення каталогу, обробки замовлень і популяризації сайту.

В економічному розділі було виконано розрахунки витрат на розробку програмного продукту, визначено собівартість, ціну роботи, економічну ефективність і термін окупності. У розділі з охорони праці розглянуто заходи щодо запобігання травматизму та професійних захворювань, а також визначено параметри мікроклімату для безпечних умов роботи.

Отже, мету кваліфікаційної роботи досягнуто. Було створено web-сайт інтернет-магазину «Лісова Майстерня», який забезпечує перегляд каталогу товарів, оформлення замовлень, подання індивідуальних заявок, авторизацію користувачів, роботу профілю та адміністрування магазину. Розроблений програмний продукт може бути використаний як навчальний приклад або як основа для подальшого розвитку реального сайту майстерні.

У майбутньому сайт можна вдосконалити шляхом додавання онлайн-оплати, системи відгуків, розширеної фільтрації товарів, автоматичного розрахунку вартості індивідуального замовлення та інтеграції з сервісами доставки.

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						91
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ПОСИЛАНЬ

1. Марціяш Г.Я., Слободян Р.О. Методичні вказівки до виконання кваліфікаційної роботи для здобувачів фахової передвищої освіти спеціальності 122 «Комп'ютерні науки». Тернопіль : ВСП «ТФК ТНТУ ім. І. Пулюя».
2. REM Online. Система обліку клієнтів, замовлень і сервісних робіт : вебсайт. URL: <https://remonline.app/> (дата звернення: 12.04.2026).
3. React. Creating a React App : вебсайт. URL: <https://react.dev/learn/creating-a-react-app> (дата звернення: 15.04.2026).
4. Vite. Next Generation Frontend Tooling : вебсайт. URL: <https://vite.dev/> (дата звернення: 18.04.2026).
5. Spring Boot. Documentation Overview : вебсайт. URL: <https://docs.spring.io/spring-boot/documentation.html> (дата звернення: 15.04.2026).
6. PostgreSQL. Documentation : вебсайт. URL: <https://www.postgresql.org/docs/> (дата звернення: 17.04.2026).
7. Spring Data JPA. Reference Documentation : вебсайт. URL: <https://docs.spring.io/spring-data/jpa/reference/index.html> (дата звернення: 16.04.2026).
8. Axios. Promise based HTTP client for the browser and node.js : вебсайт. URL: <https://axios-http.com/> (дата звернення: 19.04.2026).
9. Spring Security. Reference Documentation : вебсайт. URL: <https://docs.spring.io/spring-security/reference/index.html> (дата звернення: 16.04.2026).
10. diagrams.net (draw.io). Онлайн-сервіс для створення структурних схем та блок-діаграм : вебсайт. URL: <https://app.diagrams.net/> (дата звернення: 24.05.2026).
11. Canva. Онлайн-платформа для створення графічних матеріалів, схем та презентацій : вебсайт. URL: <https://www.canva.com/> (дата звернення: 25.05.2026).
12. W3Schools. Навчальний ресурс з web-технологій HTML, CSS та

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						92
Зм.	Арк.	№ докум.	Підпис	Дата		

JavaScript : вебсайт. URL: <https://www.w3schools.com/> (дата звернення: 14.04.2026).

13. Vercel. Documentation : вебсайт. URL: <https://vercel.com/docs> (дата звернення: 02.06.2026).

14. Render. Deploying on Render : вебсайт. URL: <https://render.com/docs/deloys> (дата звернення: 02.06.2026).

15. Node.js. Open-source, cross-platform JavaScript runtime environment : вебсайт. URL: <https://nodejs.org/en> (дата звернення: 18.04.2026).

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						93
Зм.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

Додаток А. App.jsx

```
<Routes>
  <Route path="/" element={<HomePage canOrder={!userRole} />} />
  <Route path="/login" element={<Login setUserRole={setUserRole} />} />
  <Route path="/register" element={<Register />} />
  <Route path="/forgot-password" element={<ForgotPassword />} />
  <Route path="/reset-password" element={<ResetPassword />} />
  <Route path="/products" element={<ProductList canOrder={!userRole} />}
/>
  <Route path="/products/:id" element={<ProductDetail canOrder={!userRole}
/>} />
  <Route
    path="/regular-products"
    element={<Navigate
to="/products?view=inStock" replace />} />
  <Route path="/custom-order" element={userRole ? <CustomOrderPage /> :
<Navigate to="/login" replace />} />
  <Route
    path="/profile"
    element={
      userRole ? <UserProfile /> : <Navigate to="/login" replace />
    }
  />

  <Route
    path="/admin/products"
    element={
      <ProtectedRoute role={userRole} allowedRoles={['admin']}>
        <AdminProductList />
      </ProtectedRoute>
    }
  />
  <Route
    path="/admin/users"
```

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						94
Зм.	Арк.	№ докум.	Підпис	Дата		

```

    element={
      <ProtectedRoute role={userRole} allowedRoles={['admin']}>
        <AdminUserPage />
      </ProtectedRoute>
    }
  />
<Route
  path="/admin/orders"
  element={
    <ProtectedRoute role={userRole}>
      <AdminOrdersUnifiedPage />
    </ProtectedRoute>
  }
/>
</Routes>

const ProtectedRoute = ({ children, role, allowedRoles = ['admin',
'manager'] }) => {
  return allowedRoles.includes(role) ? children : <Navigate to="/login" />;
};

```

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
						95
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток Б. ProductList.jsx

```
<div className="products-page">
  <div className="products-header">
    <h2>{title}</h2>
    <p>{subtitle}</p>
  </div>

  <div className="catalog-tabs catalog-tabs--horizontal" role="tablist">
    <button
      type="button"
      role="tab"
      aria-selected={catalogTab === 'all'}
      className={`catalog-tab ${catalogTab === 'all' ? 'catalog-tab--
active' : ''}`}
      onClick={() => setCatalogTab('all')}
    >
      Усі товари
    </button>
    <button
      type="button"
      role="tab"
      aria-selected={catalogTab === 'inStock'}
      className={`catalog-tab ${catalogTab === 'inStock' ? 'catalog-tab--
active' : ''}`}
      onClick={() => setCatalogTab('inStock')}
    >
      У наявності
    </button>
  </div>

  <div className="products-controls">
    <div className="search-bar">
      <input
        type="text"

```

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						96
Зм.	Арк.	№ докум.	Підпис	Дата		


```

        placeholder="Пошук (на сервері): назва, опис, категорія..."
        value={searchInput}
        onChange={(e) => setSearchInput(e.target.value)}
    />
</div>

{categories.length > 0 && (
    <div className="sort-options">
        <label htmlFor="pl-cat">Категорія:</label>
        <select
            id="pl-cat"
            value={category}
            onChange={(e) => setCategory(e.target.value)}
        >
            <option value="">Усі</option>
            {categories.map((c) => (
                <option key={c} value={c}>
                    {c}
                </option>
            ))}
        </select>
    </div>
)}
</div>
</div>

```

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		97

Додаток В. config.js

```
import axios from 'axios';

const API_URL = import.meta.env.VITE_API_URL || 'http://localhost:8081';

const api = axios.create({
  baseURL: API_URL,
  headers: {
    'Content-Type': 'application/json'
  },
  withCredentials: true
});

api.interceptors.request.use((config) => {
  const token = localStorage.getItem('token');
  if (token) {
    config.headers.Authorization = `Bearer ${token}`;
  }
  return config;
});

export default api;
```

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
						98
Зм.	Арк.	№ докум.	Підпис	Дата		

Додаток Г. UserController.java

```
@PostMapping("/register")
public Map<String, Object> register(HttpServletRequest request,
@RequestBody Map<String, Object> body) {
    String name = String.valueOf(body.getOrDefault("name",
body.getOrDefault("firstName", ""))).trim();
    String lastName = String.valueOf(body.getOrDefault("lastName",
"")).trim();
    String password = String.valueOf(body.getOrDefault("password",
"")).trim();
    String email = String.valueOf(body.getOrDefault("email",
"")).trim().toLowerCase();
    String phone = String.valueOf(body.getOrDefault("phone", "")).trim();
    enforceRateLimit("register", clientKey(request, email + phone), 8, 10 *
60 * 1000L);
    if (name.isEmpty() || password.isEmpty() || (email.isEmpty() &&
phone.isEmpty())) {
        throw new ResponseStatusException(HttpStatus.BAD_REQUEST, "Вкажіть
ім'я, пароль і email або телефон");
    }
    if (!email.isEmpty() && userRepository.findByEmail(email).isPresent())
{
        throw new ResponseStatusException(HttpStatus.BAD_REQUEST,
"Користувач із таким email вже існує");
    }
    if (!phone.isEmpty() && userRepository.findByPhone(phone).isPresent())
{
        throw new ResponseStatusException(HttpStatus.BAD_REQUEST,
"Користувач із таким телефоном вже існує");
    }
    UserEntity u = new UserEntity();
    u.setId(idService.nextId());
    u.setName(name);
    u.setLastName(lastName.isBlank() ? null : lastName);
}
```

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		99

```

        u.setPassword(encoder.encode(password));
        u.setRole("client");
        u.setActive(true);
        u.setEmail(email.isEmpty() ? null : email);
        u.setPhone(phone.isEmpty() ? null : phone);
        userRepository.save(u);
        return Map.of("message", "Реєстрація успішна");
    }

    @PostMapping("/login")
    public Map<String, Object> login(HttpServletRequest request, @RequestBody
    Map<String, Object> body) {
        String login = String.valueOf(body.getOrDefault("login",
        body.getOrDefault("email", ""))).trim().toLowerCase();
        String password = String.valueOf(body.getOrDefault("password",
        "")).trim();
        enforceRateLimit("login", clientKey(request, login), 12, 10 * 60 *
        1000L);
        UserEntity user = login.contains("@")
            ? userRepository.findByEmail(login).orElse(null)
            : userRepository.findByPhone(login).orElse(null);
        if (user == null || !encoder.matches(password, user.getPassword())) {
            throw new ResponseStatusException(HttpStatus.BAD_REQUEST, "Невірні
            облікові дані");
        }
        String token = jwtService.generateToken(user.getId(),
        publicRole(user.getRole()), user.getEmail(), user.getPhone());
        return Map.of("token", token, "role", publicRole(user.getRole()));
    }

```

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		100

Додаток Д. ProductController.java

@PostMapping

```
public Map<String, Object> create(HttpServletRequest request, @RequestBody
Map<String, Object> body) {
    RequestUser user = WebSupport.requireUser(request);
    WebSupport.requireAdmin(user);
    ProductEntity p = fromBody(new ProductEntity(), body);
    p.setId(idService.nextId());
    productRepository.save(p);
    return Map.of("message", "Товар додано", "product", toDto(p));
}
```

@PutMapping("/{id}")

```
public Map<String, Object> update(HttpServletRequest request, @PathVariable
String id, @RequestBody Map<String, Object> body) {
    RequestUser user = WebSupport.requireUser(request);
    WebSupport.requireAdmin(user);
    ProductEntity p = productRepository.findById(id).orElseThrow(() -> new
ResponseStatusException(HttpStatus.NOT_FOUND, "Товар не знайдено"));
    fromBody(p, body);
    productRepository.save(p);
    return Map.of("message", "Товар оновлено", "product", toDto(p));
}
```

@DeleteMapping("/{id}")

```
public Map<String, Object> delete(HttpServletRequest request, @PathVariable
String id) {
    RequestUser user = WebSupport.requireUser(request);
    WebSupport.requireAdmin(user);
    if (!productRepository.existsById(id)) throw new
ResponseStatusException(HttpStatus.NOT_FOUND, "Товар не знайдено");
    productRepository.deleteById(id);
    return Map.of("message", "Товар видалено");
}
```

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		101

Додаток Е. CustomOrderController.java

```
@PatchMapping("/{orderId}/status")
public Map<String, Object> updateStatus(HttpServletRequest request,
@PathVariable String orderId, @RequestBody Map<String, Object> body) {
    RequestUser user = WebSupport.requireUser(request);
    WebSupport.requireAdminOrManager(user);
    OrderEntity order = orderRepository.findById(orderId).orElseThrow(() ->
new    ResponseStatusException(HttpStatus.NOT_FOUND,    "Замовлення не
знайдено"));
    String    to    =    String.valueOf(body.getOrDefault("status",
order.getStatus()));
    appendTimeline(order,    Map.of("kind",    "status_change",    "at",
Instant.now().toString(),    "actorName",    user.name(),    "fromStatus",
order.getStatus(), "toStatus", to));
    order.setStatus(to);
    orderRepository.save(order);
    return Map.of("success", true, "message", "Замовлення оновлено",
"order", toOrderDto(order));
}
```

```
@PostMapping
public Map<String, Object> create(HttpServletRequest request, @RequestBody
Map<String, Object> body) {
    RequestUser user = WebSupport.requireUser(request);
    Map<String,    Object>    params    =
WebSupport.asMap(body.getOrDefault("params", Map.of()));
    Object referenceUrls = body.getOrDefault("referenceUrls", List.of());
    int referenceCount = referenceUrls instanceof Collection<?> collection
? collection.size() : 0;
    CustomPriceCalculator.Estimate estimate;
    try {
        estimate = CustomPriceCalculator.calculate(params, referenceCount);
    } catch (IllegalArgumentException ex) {
        throw new ResponseStatusException(HttpStatus.BAD_REQUEST, "Вкажіть
```

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		102

```

    коректні розміри для розрахунку вартості");
    }

    CustomOrderEntity e = new CustomOrderEntity();
    e.setId(idService.nextId());
    e.setOrderNumber(orderNumberService.next());
    e.setUserId(user.id());
    e.setContactJson(jsonService.write(body.getOrDefault("contact",
Map.of()))));
    e.setParamsJson(jsonService.write(params));
    e.setReferenceUrlsJson(jsonService.write(referenceUrls));
    e.setEstimatedPrice(estimate.total());
    customOrderRepository.save(e);
    return Map.of("success", true, "customOrder", toDto(e));
}

```

					2026.КВР.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		103

Додаток Ж. AuthFilter.java

```
protected void doFilterInternal(HttpServletRequest request,
HttpServletResponse response, FilterChain filterChain) throws
ServletException, IOException {
    String auth = request.getHeader("Authorization");
    if (auth == null || !auth.startsWith("Bearer ")) {
        filterChain.doFilter(request, response);
        return;
    }

    try {
        Map<String, Object> claims = jwtService.parse(auth.substring(7));
        String userId = String.valueOf(claims.get("id"));
        UserEntity user = userRepository.findById(userId).orElse(null);
        if (user == null || !user.isActive()) {
            SecurityContextHolder.clearContext();
            response.sendError(HttpServletResponse.SC_FORBIDDEN, "Account
is inactive or not found");
            return;
        }

        RequestUser requestUser = new RequestUser(
            user.getId(),
            user.getRole(),
            user.getEmail(),
            user.getPhone(),
            user.getName()
        );
        request.setAttribute("requestUser", requestUser);

        String role = String.valueOf(user.getRole() == null ? "client" :
user.getRole()).toUpperCase(Locale.ROOT);
        UsernamePasswordAuthenticationToken authentication = new
UsernamePasswordAuthenticationToken(
```

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
						104
Зм.	Арк.	№ докум.	Підпис	Дата		


```

        requestUser,
        null,
        List.of(new SimpleGrantedAuthority("ROLE_" + role))
    );
    authentication.setDetails(new
WebAuthenticationDetailsSource().buildDetails(request));

SecurityContextHolder.getContext().setAuthentication(authentication);

    filterChain.doFilter(request, response);
} catch (Exception ex) {
    SecurityContextHolder.clearContext();
    response.sendError(HttpServletResponse.SC_UNAUTHORIZED, "Invalid
token");
}
}

```

					2026.KBP.122.423.03.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		105