

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Навчально-науковий інститут новітніх освітніх технологій
Кафедра інформаційно-обчислювальних систем і управління

КОТРУБЕНКО Валерія Ігорівна

Модуль виконання замовлень в онлайн-магазині прикрас / Order fulfillment module in an online jewelry store

спеціальність: 122 - Комп'ютерні науки
освітньо-професійна програма - Комп'ютерні науки

Кваліфікаційна робота

Виконала студентка групи КНз-41
В.І. Котрубенко

Науковий керівник:
к.т.н., доцент Н.М. Васильків

Кваліфікаційну роботу
допущено до захисту:
«___» _____ 20___ р.
В.о. завідувача кафедри
_____ Н.В. Дзюбановська

ТЕРНОПІЛЬ – 2026

Навчально-науковий інститут новітніх освітніх технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність: 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Н.В. Дзюбановська
«_____» _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

КОТРУБЕНКО Валерії Ігорівні

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи **Модуль виконання замовлень в онлайн-магазині прикрас / Order fulfillment module in an online jewelry store**

керівник роботи к.т.н., доцент Н.М. Васильків

затверджений наказом по університету від 01 грудня 2025 року № 894.

2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2026 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити

– провести аналіз предметної області, вимог користувачів та дослідити існуючі аналоги на ринку;

– сформулювати технічні та функціональні вимоги до програмного модуля;

– спроектувати архітектуру бази даних та логіку взаємодії клієнтської і серверної частин додатка;

– розробити адаптивні компоненти користувацького інтерфейсу в лаконічному візуальному стилі;

– програмно реалізувати серверні сервіси для обробки операцій та керування контентом;

– провести тестування розробленого модуля на предмет коректності роботи та швидкодії.

5. Перелік графічного матеріалу у роботі

- модель бази даних;

- схема алгоритму функціонування модуля

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01 грудня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Огляд літературних джерел відповідно до предметної області та складання плану роботи	до 01.01.2026 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02. 2026 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 15.03.2026 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05.2026 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2026 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2026 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2026 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2026 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06.2026 р.	

Студент _____ В.І. Котрубенко
підпис

Керівник роботи _____ к.т.н., доцент Н.М. Васильків
підпис

АНОТАЦІЯ

Кваліфікаційна робота на тему «Модуль виконання замовлень в онлайн-магазині прикрас» на здобуття освітнього ступеня «бакалавр» зі спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 64 сторінки і містить 30 ілюстрацій, 9 таблиць, 3 додатки та 32 використаних джерела.

Метою роботи є проєктування та реалізація програмного модуля, що забезпечує зручний процес вибору та онлайн-замовлення прикрас.

Методами розроблення обрано метод системного аналізу, методи концептуального та логічного проєктування, об'єктно-орієнтований метод розробки.

Внаслідок виконання роботи обґрунтовано архітектуру бази даних, реалізовано програмні компоненти взаємодії з користувачем та розроблено програмний засіб виконання замовлень, який забезпечує створення готового онлайн-магазину прикрас.

Результати дослідження можуть бути використані приватними підприємцями та ювелірними брендами для власних торгових.

Ключові слова: ОНЛАЙН-МАГАЗИН, ЮВЕЛІРНІ ПРИКРАСИ, МОДУЛЬ ВИКОНАННЯ ЗАМОВЛЕНЬ, БАЗА ДАНИХ, ІНТЕРФЕЙС КОРИСТУВАЧА.

ANNOTATION

The qualification work on the topic " Order fulfillment module in an online jewelry store " for the bachelor's degree in specialty 122 "Computer Science", educational program "Computer Science" consists of 64 pages and contains 30 illustrations, 9 tables, 3 appendices, and 32 references.

The goal of this project is to design and implementation of a software module that provides a convenient process of selecting and online ordering of jewelry.

The development methods include the method of systems analysis, methods of conceptual and logical design, and the object-oriented development method.

As a result of the work, the database architecture has been justified, software components for user interaction have been implemented, and an order fulfillment software tool has been developed, ensuring the creation of a deployment-ready commercial online jewelry store.

The research results can be used by private entrepreneurs and jewelry brands to deploy their own e-commerce web platforms.

Keywords: ONLINE STORE, JEWELRY, ORDER FULFILLMENT MODULE, DATABASE, USER INTERFACE

ЗМІСТ

Вступ.....	7
1 Аналіз предметної області та обґрунтування автоматизації процесів електронної комерції	9
1.1 Аналіз предметної області.....	9
1.2 Огляд і аналіз існуючих аналогів	11
1.3 Постановка задачі дослідження.....	16
2 Алгоритмічне та інформаційне забезпечення модуля.....	18
2.1 Проектування структури модуля	18
2.2 Алгоритм функціонування модуля.....	22
2.3 Інформаційне забезпечення.....	24
3 Програмно-технологічне забезпечення.....	344
3.1 Реалізація бази даних	344
3.2 Інтерфейс користувача.....	40
3.3 Тестування модуля	43
Висновки	477
Список використаних джерел	488
Додаток А. Лістинги схем бази даних (моделей MongoDB)	511
Додаток Б. Програмний код модуля замовлень	524
Додаток В. Копія публікації.....	59

ВСТУП

Сучасний бізнес стрімко масштабується в інтернет-просторі. Цифрові продажі зараз виступають головним рушієм комерційного зростання. Ринок ювелірних виробів займає вагомий нішу у цій новій економіці.

Задача, яка виникла на об'єкті проєктування, пов'язана з необхідністю створення швидких та інтуїтивно зрозумілих інструментів для онлайн-торгівлі. Малий бізнес та локальні бренди потребують гнучкого рішення, яке дозволить запустити платформу з мінімальними витратами. Проте проблемою залишається технічна реалізація процесу замовлення, яка повинна враховувати специфіку ювелірної сфери, варіативність характеристик товарів та коректний збір даних. Тому створення спеціалізованого модуля виконання онлайн-замовлень ювелірних прикрас, що інтегрує каталог та кошик, є раціональним кроком для вирішення цих завдань.

Об'єктом дослідження є процеси електронної комерції, спрямовані на реалізацію ювелірних виробів через веб-платформи, а також логістичні цикли обробки заявок користувачів.

Предмет дослідження - автоматизація виконання замовлень в онлайн-магазині прикрас.

Метою роботи є проєктування та реалізація програмного модуля, що забезпечує зручний процес вибору та онлайн-замовлення прикрас.

Для досягнення мети проаналізовано особливості інтернет-ринку прикрас, переваги та недоліки існуючих програмних рішень у цій сфері діяльності.

Центральним елементом модуля є база даних продукції, яка містить детальні описи, ціни та посилання на зображення виробів, забезпечуючи зручність при перегляді каталогу. Логіка завершення покупки реалізована через форму збору контактних даних, після заповнення якої програмний код формує об'єкт замовлення та відправляє його на пошту менеджера. Весь процес супроводжується валідацією введених даних, щоб гарантувати коректність зв'язку з покупцем та точність інформації про обрані ювелірні вироби.

Програмний модуль є повнофункціональним та автоматизовує весь цикл обробки замовлень. Функціональність включає інтерактивний каталог товарів, а також динамічний кошик з автоматичним підрахунком вартості.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОБҐРУНТУВАННЯ АВТОМАТИЗАЦІЇ ПРОЦЕСІВ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

1.1 Аналіз предметної області

Ювелірні вироби належать до специфічної категорії товарів, які поєднують у собі матеріальну цінність дорогоцінних металів – золота, срібла чи платини, а також естетичну привабливість вставок із коштовного або напівкоштовного каміння. Розробка інформаційної системи для такої ніші базується на трьох ключових факторах: високій вартості одиниці товару, унікальності технічних параметрів кожного виробу та важливості якісної візуалізації.

Сучасний ринок електронної комерції вимагає особливого підходу до реалізації ювелірної продукції. Станом на жовтень 2025 року в Україні зареєстровано 622 юридичні особи та 855 ФОПів, для яких виробництво ювелірних виробів є основним видом економічної діяльності. Переважна більшість (понад 32%) традиційно зосереджена в Києві (200 компаній), а також на Харківщині (79) та Вінниччині (51) [30].

Аналіз ринку демонструє стабільну трансформацію індустрії навіть у складні воєнні часи: зростає попит на унікальний дизайн, індивідуальні замовлення та використання перероблених металів, а також спостерігається цифровізація продажу.

Онлайн-замовлення прикрас в Україні — це розвинений цифровий досвід, який поєднує зручність маркетплейсів із гарантіями безпеки. Сучасні платформи максимально мінімізують ризики дистанційної покупки.

Український сегмент інтернет-комерції ювелірних виробів представлений мультибрендовими гігантами та мережами («Укрзолото», «Золотий Вік»), дизайнерськими та трендовими брендами («Oniks», «Pandora»), а також індивідуальними онлайн-замовленнями на вироби.

Технологія доповненої реальності в онлайн-покупках вивела споживчий досвід в ювелірній сфері на новий рівень завдяки віртуальним приміркам і персоналізованим пропозиціям.

Просування ювелірного інтернет-магазину передбачає цілий комплекс дій: від структурування каталогу до створення текстів, які привертають увагу і допомагають людині зробити вибір. Особливе значення надається високоякісному візуальному контенту — зображення мають максимально точно передавати текстуру прикрас, глибину блиску каміння та форму виробу. Візуальне сприйняття значною мірою визначає цінність продукту та впливає на рішення клієнта придбати виріб. Необхідно враховувати і сезонні коливання попиту, наприклад святкові періоди, модні тренди тощо. Це потребує гнучкої маркетингової стратегії, своєчасного коригування рекламних активностей і продуманого планування просування з урахуванням прогнозованих піків і спадів інтересу [29].

Для підвищення залученості та зміцнення лояльності до бренду доцільно надавати клієнтам матеріали про догляд за прикрасами, пропонувати вибір виробів до важливих подій і символіку каміння.

На рівень продажів впливають також і технічні параметри сайту — швидкість, стабільність і коректність відображення контенту.

Просування ювелірного інтернет-магазину неможливе без детально опрацьованих описів товарів і якісного візуального контенту, який не лише естетично привабливий, а й інформативно розкриває особливості виробу, підсилюючи довіру та стимулюючи придбання виробу [29]:

- кожна прикраса має мати окрему оптимізовану сторінку з фотоматеріалами з різних ракурсів, детальним описом характеристик, зазначенням матеріалів, розмірів, ваги та можливих стилістичних поєднань з іншими виробами;
- категорії товарів (кільця, ланцюжки, браслети, сережки) мають бути логічно структуровані та інтуїтивно зрозумілі;
- ціни, акції та спеціальні пропозиції розміщуються в зонах високої видимості та легко сприймаються з першого погляду;
- кнопки «Купити» та «Додати до кошика» мають бути чітко виділені візуально та залишатися доступними на всіх етапах взаємодії зі сторінкою.

Коли магазин оновлюється, наповнюється корисними матеріалами, чесно веде комунікацію та працює з аналітикою, він буде актуальним.

1.2 Огляд і аналіз існуючих аналогів

Об'єктами аналізу стали світові бренди Pandora (рисунк 1.1) та Tiffany & Co (рисунк 1.2), а також українська мережа «Золотий Вік» (рисунк 1.3). Кожна з цих платформ використовує власні підходи для взаємодії з клієнтом, які слугують базою для визначення технічних вимог до розробки модуля.

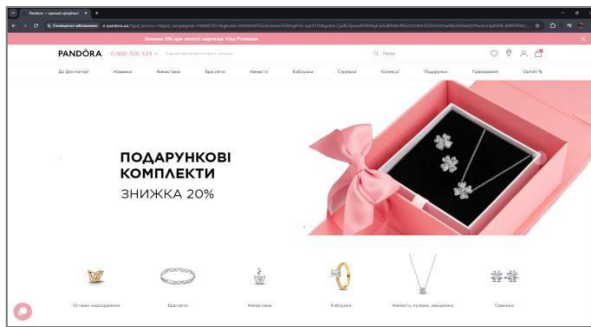


Рисунок 1.1 – Інтерфейс сайту Pandora

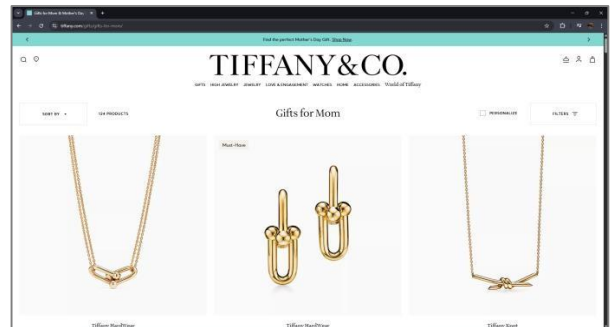


Рисунок 1.2 – Інтерфейс сайту Tiffany & Co

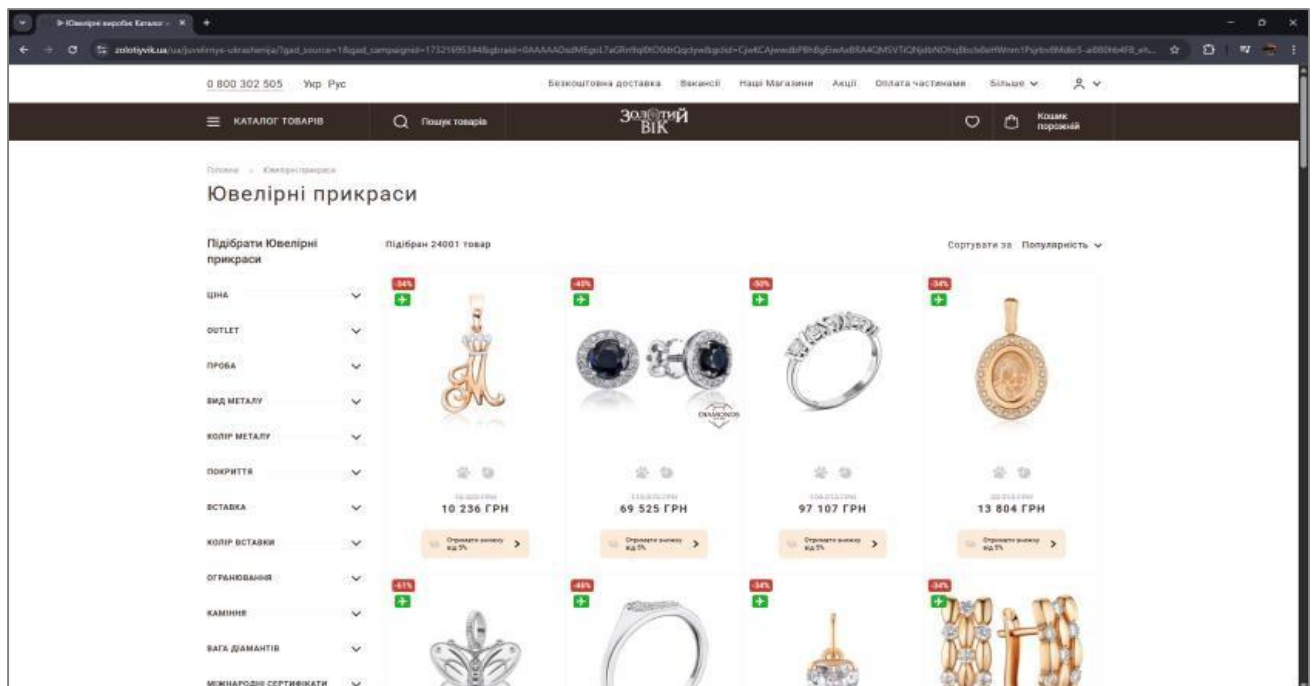


Рисунок 1.3 – Сайт Золотий Вік

Сайт бренду Pandora [25] робить акцент на глибокій персоналізації товарів. Головною перевагою платформи є продуманий механізм вибору специфічних параметрів виробу (рисунк 1.4). Інтерфейс забезпечує користувача чіткими візуальними підказками для правильного визначення розмірів, що є критично

важливим для ювелірного сегмента (рисунок 1.5).

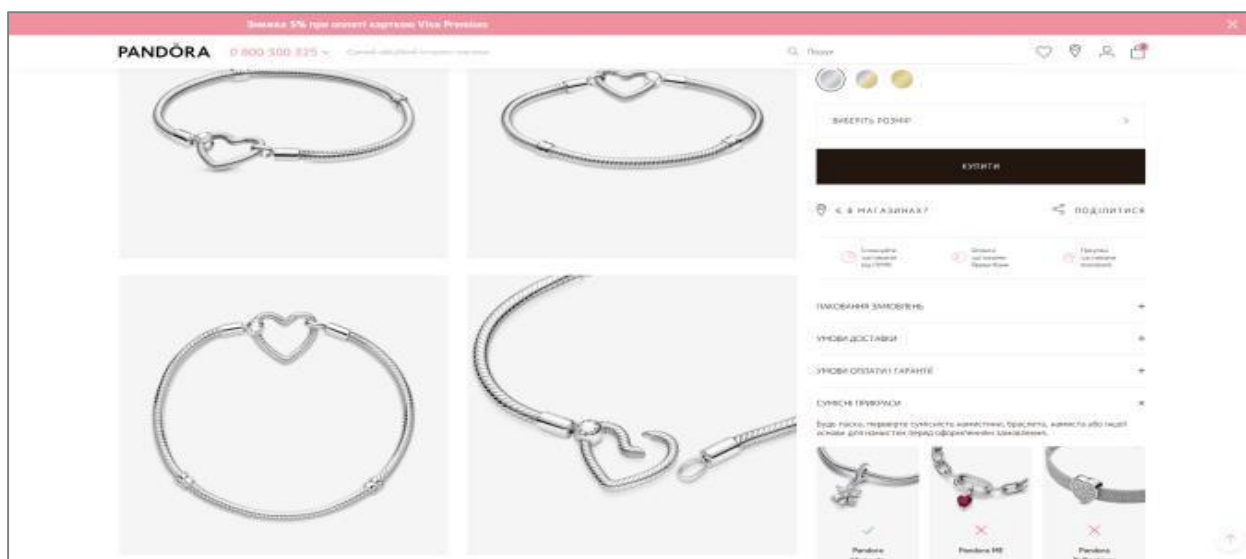


Рисунок 1.4 - Вибір кольору виробу

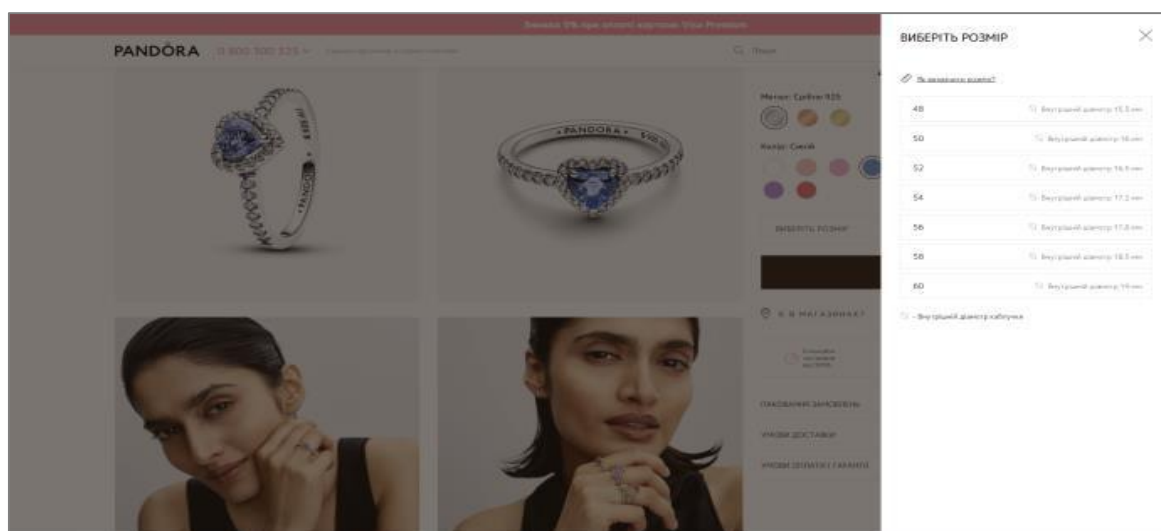


Рисунок 1.5 – Вибір розміру та кольору каблучки

Однак під час дослідження платформи виявлено низку недоліків. Оскільки асортимент переважно складається з базових браслетів та великої кількості дрібних аксесуарів до них, сторінка каталогу виглядає візуально перенасиченою. Це ускладнює швидкий пошук та фільтрацію потрібної позиції. Додатковою проблемою є агресивний підхід до додаткових продажів: автоматичне додавання у кошик супутніх товарів створює ефект нав'язливого маркетингу (рисунок 1.6), що знижує загальний комфорт від покупки.

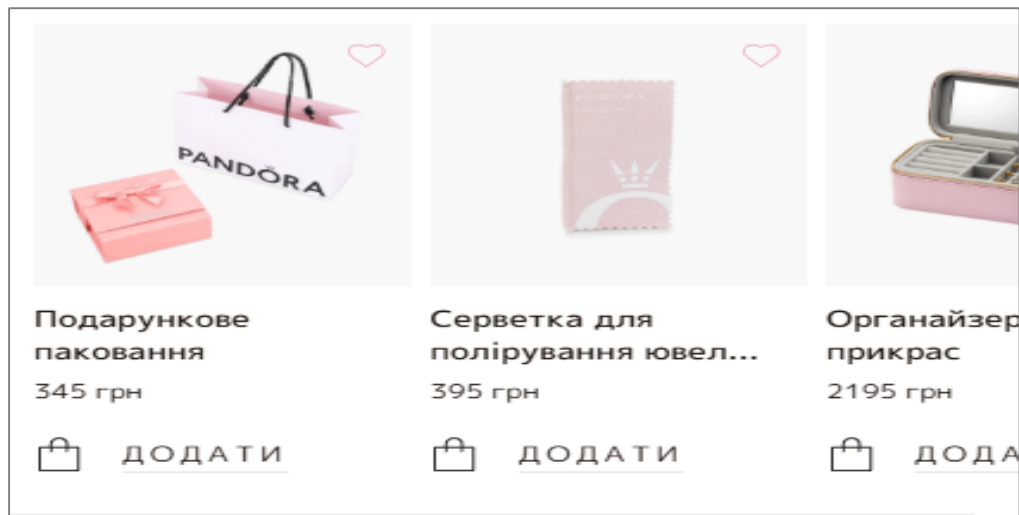


Рисунок 1.6 – Додаткові пропозиції

Вебплатформа Tiffany & Co [26] концептуально відрізняється та орієнтується на суворий мінімалізм і візуальну преміальність (рисунок 1.7). Взаємодія з клієнтом будується довкола високоякісного візуального контенту та максимально спрощеного процесу оформлення замовлення, який не перевантажує покупця зайвими кроками.

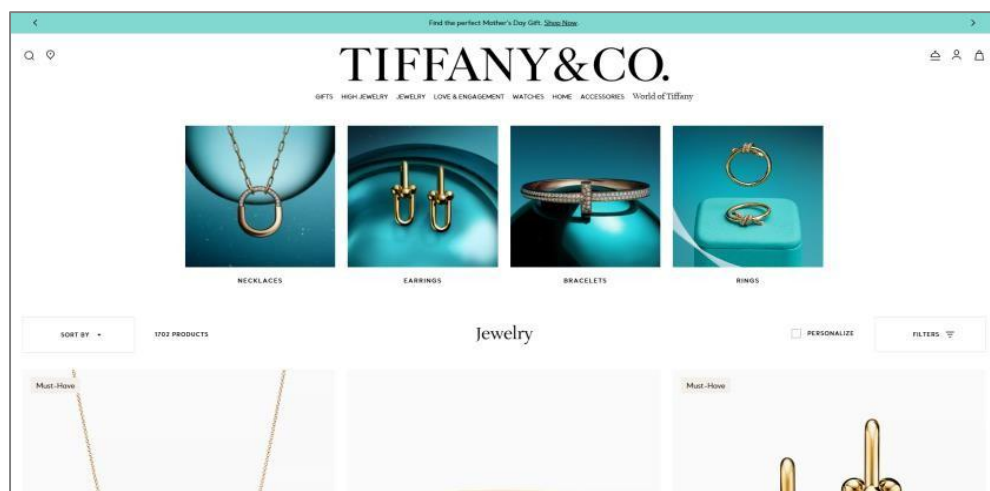


Рисунок 1.7 – Інтерфейс сайту Tiffany & Co

Головний недолік цього рішення полягає у регіональних обмеженнях: повноцінний функціонал електронної комерції доступний лише для певного переліку країн. Користувачі з інших регіонів можуть лише переглядати каталог без доступу до актуальних цін чи можливості онлайн-покупки. Також аналіз

клієнтських відгуків [27] за 2025–2026 роки вказує на наявність проблем із синхронізацією статусів. Інформація про доставку або сервісне обслуговування часто не відповідає реальному стану обробки запиту (рисунки 1.8, 1.9), що неприпустимо для систем такого рівня.



Рисунок 1.8 – Відгук про ремонт

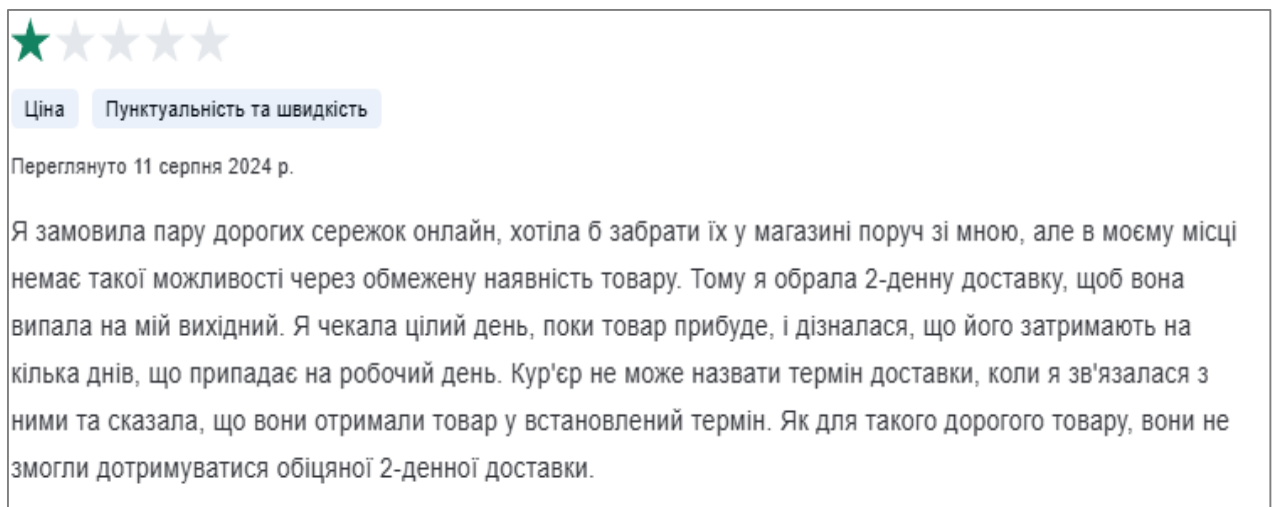


Рисунок 1.9 – Відгук про затримку відправки замовлення

Український онлайн-магазин «Золотий Вік» [28] демонструє потужний технічний підхід до роботи з базою даних. Платформа пропонує розгалужену систему фільтрації за кольором та пробою дорогоцінного металу (рисунок 1.10), що свідчить про високий рівень деталізації товарного каталогу.

маркетинговий функціонал або важкі багаторівневі фільтри не притягують увагу користувача. Для проєкту доцільно обрати лаконічну візуальну концепцію, яка сфокусує увагу клієнта безпосередньо на виробі та забезпечить, швидкий процес оформлення замовлення із можливістю гнучкого редагування кошика.

1.3 Постановка задачі дослідження

На основі аналізу предметної області та недоліків існуючих онлайн-платформ ювелірних компаній сформовано вимоги до розроблюваного модуля виконання замовлень [8, 11]. Головною задачею є створення лаконічного та швидкого модуля, орієнтованого виключно на клієнтську частину, без перевантаження інтерфейсу зайвим маркетинговим функціоналом чи складними панелями управління.

Вимоги до розроблюваного програмного забезпечення поділяються на структурні, функціональні та візуальні:

1. Структурні вимоги полягають у проєктуванні гнучкої нереляційної моделі бази даних. Вона повинна динамічно пов'язувати товари з їхніми варіативними атрибутами без необхідності дублювати картки прикрас у базі. Це розв'язує проблему перевантаженості бази даних, зафіксовану в існуючих аналогах.

2. Функціональні вимоги зосереджені на реалізації клієнтської частини модуля. Головним завданням є створення динамічного кошика, у якому користувач має можливість редагувати розміри прикрас та їхню кількість перед покупкою. Також передбачається забезпечення надійної валідації контактних даних клієнта під час відправки форми замовлення.

3. Візуальні вимоги передбачають розробку інтерфейсу в стилі мінімалізму з акцентом на контент. Для забезпечення комфортної та «м'якої» взаємодії користувача з електронним каталогом візуальна концепція має базуватися на лаконічній геометрії.

Завданням дослідження є створення програмного засобу, який дозволить досягти високого рівня зручності, швидкодії та ефективності у сфері

функціонування онлайн-магазину ювелірних прикрас, надаючи готовий комерційний продукт.

Необхідно спроектувати та реалізувати програмний модуль, що автоматизує цикл «замовлення – обробка – відправка», мінімізує помилки при обліку специфічних характеристик прикрас (розміри, матеріали) та забезпечує зворотний зв'язок із покупцем.

Модуль має бути реалізований як веб-сайт з використанням сучасних фреймворків (React/Node.js). База даних повинна підтримувати цілісність даних при конкурентних запитах.

Метою роботи є проектування та реалізація програмного модуля, що забезпечує зручний процес вибору та онлайн-замовлення прикрас.

Для досягнення поставленої мети необхідно:

1. Провести аналіз предметної області, вимог користувачів та дослідити існуючі аналоги на ринку.
2. Сформулювати чіткі технічні та функціональні вимоги до програмного модуля.
3. Спроектувати архітектуру бази даних та логіку взаємодії клієнтської і серверної частин додатка.
4. Розробити адаптивні компоненти користувацького інтерфейсу в лаконічному візуальному стилі.
5. Програмно реалізувати серверні сервіси для обробки операцій та керування контентом.
6. Провести тестування розробленого модуля на предмет коректності роботи та швидкодії.

Програмний модуль має бути повнофункціональним та автоматизувати весь цикл обробки замовлень. Функціональність включає інтерактивний каталог товарів, а також динамічний кошик з автоматичним підрахунком вартості.

2 АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ МОДУЛЯ

2.1 Проектування структури модуля

Для побудови ефективної взаємодії між програмними компонентами модуля виконання замовлень в онлайн-магазині прикрас виконано концептуальне проектування його архітектури [1, 6, 9, 13]. Основною вимогою є забезпечення високої швидкості обробки клієнтських запитів та зручності під час вибору ювелірних виробів.

Розробку програмного модуля розподілено на три послідовні етапи: створення концепції, проектування структури та безпосередня функціональна реалізація (рисунок 2.1).

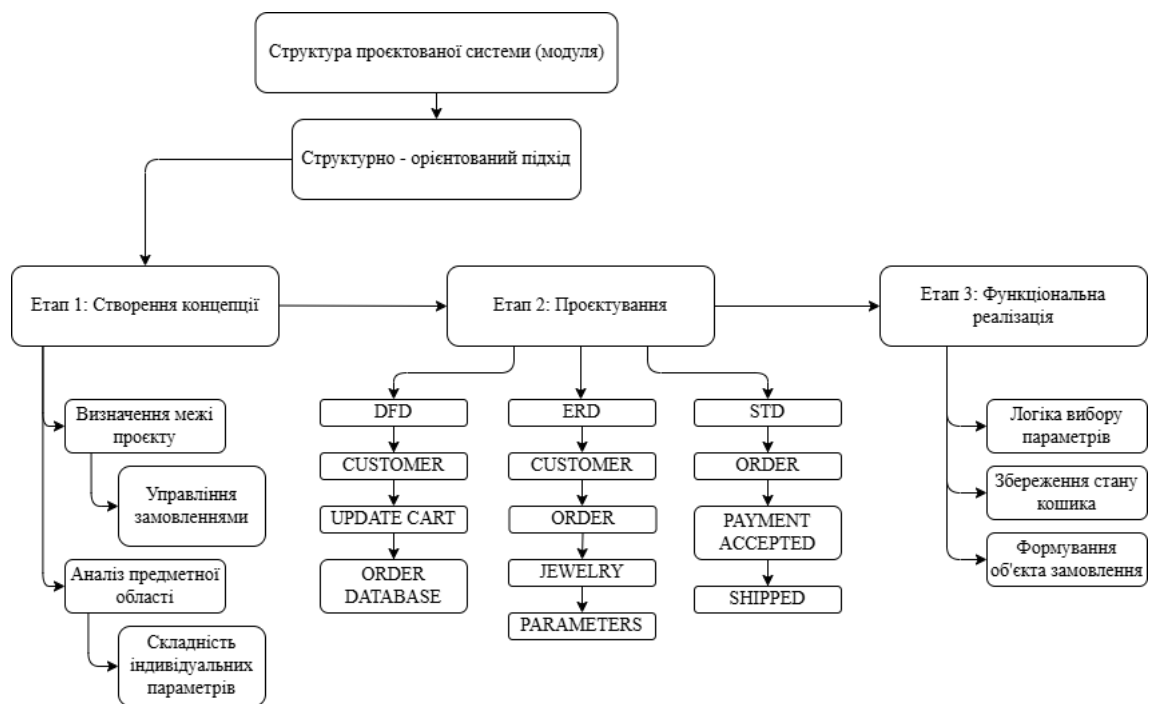


Рисунок 2.1 – Етапи створення модуля

Запропонований підхід базується на послідовній декомпозиції об'єкта, яка здійснюється шляхом структурного аналізу. Цей метод дослідження передбачає рух від загального огляду системи до її глибокої деталізації [9, 15], що дозволяє сформувати ієрархічну структуру з необхідною кількістю рівнів.

Відповідно до обраної стратегії, весь процес проектування поділено на три

основні етапи, кожен з яких відіграє ключову роль у створенні модуля:

На етапі створення концепції формується загальне бачення системи, визначаються її межі та аналізується предметна область (таблиця 2.1). Основна увага приділяється специфіці ювелірної галузі.

Таблиця 2.1 – Етап створення концепції

Етап	Складові частини	Опис та характеристики
Створення концепції	Визначення межі проєкту	Управління замовленнями дозволяє зосередитися на конкретних бізнес-цілях
	Аналіз предметної області	Аналізується специфіка ювелірної справи, особливо складність обліку індивідуальних параметрів (вага, метал, розмір), що впливає на логіку системи.

На етапі проектування використовуються діаграми потоків даних, моделей «суть-зв'язок» та діаграм переходів станів - це дозволяє спроектувати структуру бази даних та динамічну поведінку замовлення залежно від часу (таблиця 2.2).

Таблиця 2.2 – Етап проектування

Етап	Складові частини	Опис та характеристики
Проектування	DFD	Описуються інформаційні потоки: дія клієнта, оновлення кошика, та збереження даних у базі.
	ERD	Модель даних, що пов'язує клієнта, замовлення, виріб та його параметри в єдину логічну структуру.
	STD	Моделюється поведінка замовлення залежно від часу: від створення до підтвердження оплати та відправки.

На етапі реалізації (таблиця 2.3) здійснюється логіка вибору параметрів, технічні механізми збереження стану сесії кошика та фінальне формування об'єкта замовлення для подальшої обробки.

Таблиця 2.3 – Етап функціональної реалізації

Етап	Складові частини	Опис та характеристики
Функціональна реалізація	Логіка вибору параметрів	Програмне забезпечення механізму, який дозволяє користувачеві налаштувати характеристики прикраси під свої потреби.
	Збереження стану кошика	Технічна реалізація утримання даних про обрані товари протягом всієї сесії користувача на сайті.
	Формування об'єкта замовлення	Кінцевий процес збору всієї інформації в єдину базу для його подальшої обробки менеджером.

Завдяки використанню структурно-орієнтованого підходу та методу декомпозиції зверху-вниз вдалося поєднати складні процеси ювелірної галузі у визначену систему взаємопов'язаних етапів.

Об'єктом дослідження є процеси електронної комерції, які забезпечують реалізацію ювелірних виробів через спеціалізовані вебплатформи, а також цикли реєстрації та обробки замовлень від користувачів [11, 16].

Для детального моделювання структури розроблюваного модуля сформовано діаграму дерева функцій (рисунок 2.2).



Рисунок 2.2 – Дерево функцій модуля виконання замовлень

Перша гілка дерева функцій (див. рисунок 2.2) відповідає за інформаційне

забезпечення користувача, де реалізовано логіку вибору індивідуальних характеристик ювелірного виробу. Центральна гілка керує кошиком, забезпечує збереження стану сесії та надає можливість редагувати вміст списку покупок. Заключний етап відповідає за оформлення та підтвердження замовлення, забезпечуючи збір персональних і логістичних даних клієнта.

Загальну архітектуру процесу виконання онлайн-замовлень розділено на дві ключові підсистеми: транзакційну складову (збір та накопичення первинних даних у реальному часі) та аналітичну складову (обробка даних для прийняття рішень), що відображено на рисунку 2.3.



Рисунок 2.3 – Транзакційна та аналітична складова

Транзакційний блок фіксує кожну дію користувача, перевіряє коректність заповнення форм та гарантує цілісність інформації під час відправки запиту. Аналітичний блок систематизує інформацію про виконані замовлення, оцінює попит на конкретні категорії виробів і виступає основою для оптимізації асортименту інтернет-магазину.

2.2 Алгоритм функціонування модуля

Алгоритмічне забезпечення [6] є базовим компонентом модуля, який об'єднує сукупність математичних методів, моделей та обчислювальних алгоритмів для перетворення вхідних даних у вихідні показники. Головне завдання цього блоку полягає в організації логіки обробки подій кошика та фіналізації замовлення, що схематично зображено на рисунку 2.4.



Рисунок 2.4 – Узагальнена схема алгоритму функціонування модуля

Процес взаємодії користувача з платформою починається з каталогу товарів. Відвідувач використовує систему фільтрації, щоб відсортувати прикраси за типом металу, пробою, вагою або типом вставок.

Фільтрація дозволяє швидко знайти необхідну модель серед широкого асортименту, що є критично важливим для утримання клієнтської уваги. Система автоматично оновлює результати видачі, адаптуючись до запитів користувача, що забезпечує безперервну динаміку перегляду.

Після вибору моделі, клієнт переходить до етапу конфігурації. Унікальність ювелірної сфери полягає в тому, що багато виробів потребують індивідуального налаштування параметрів, наприклад, вибору розміру кільця, типу вставки, довжини ланцюжка або браслета тощо.

Програмний модуль у реальному часі проводить валідацію – технічну перевірку коректності обраних характеристик. Паралельно з цим виконується математичний розрахунок вартості виробу, враховуючи всі змінні параметри, що гарантує цінову прозорість для покупця.

Наступним кроком є формування замовлення в кошику. Користувач додає товари у структуру віртуального кошика, де зберігаються всі визначені атрибути прикрас. Цей етап слугує проміжним буфером, що дозволяє клієнту переглянути список покупок перед остаточним рішенням. Модуль кошика забезпечує цілісність даних, передаючи повний набір характеристик кожного виробу до наступного етапу обробки без втрати параметрів.

Після завершення базових обчислювальних операцій алгоритм ініціює передачу отриманих результатів до блоку управління сесією кошика. На цьому етапі відбувається фіксація вибору користувача та формування структурованого списку покупок.

Фінальна стадія роботи проєктованого модуля полягає в оформленні замовлення та перетворенні вибору користувача на транзакцію. Клієнт вносить контактні дані, визначає спосіб доставки та метод оплати. Після підтвердження операції алгоритм ініціює запис даних до бази керування замовленнями. Таким чином, система завершує цикл обробки, автоматично створюючи фіксований запис

про продаж та повідомляючи користувача про успішне виконання операції. Такий багаторівневий підхід мінімізує ризики помилок під час формування замовлень та забезпечує надійність фінансових розрахунків.

Завершальна стадія роботи алгоритму, функціональні характеристики якого наведені в таблиці 2.4, орієнтована на забезпечення технічної точності розрахунків.

Таблиця 2.4 – Функціональна характеристика алгоритму обробки даних

Етап обробки даних	Зміст та характеристика операції	Логічний зв'язок з наступним етапом
Ініціалізація запиту	Збір параметрів виробу та ідентифікація сесії користувача	Створює масив вхідних даних для математичної моделі
Математична обробка	Розрахунок підсумкової вартості за ціновою формулою	Передає верифікований фінансовий результат у кошик
Валідація та контроль	Перевірка коректності заповнення форми та контактних даних	Дозволяє або блокує доступ до фінального запису в БД
Транзакційна фіналізація	Реєстрація об'єкта замовлення та очищення тимчасової пам'яті	Оновлює стан бази даних та завершує цикл обслуговування

Достовірність вихідних результатів гарантується механізмом транзакційної цілісності: запис сформованого об'єкта до бази даних виконується виключно після успішного проходження всіх попередніх етапів обробки без виникнення системних збоїв.

2.3 Інформаційне забезпечення

Інформаційне забезпечення – це комплекс методів організації, класифікації та структурування масивів даних, що використовуються в автоматизованій системі для реалізації її функціональних завдань [8]. У межах розроблюваного модуля виконання замовлень ювелірних виробів, інформаційне забезпечення організовує архітектуру даних, яка забезпечує цілісність, системну доступність та поточний стан кошика користувача.

Проектування цієї складової базується на логічній інтеграції внутрішніх сховищ даних із зовнішніми запитами (таблиця 2.5), що надходять від клієнтського інтерфейсу. Відповідно до принципів системного аналізу, структура даних мінімізує інформаційну надлишковість та оптимізує швидкість виконання операцій читання й запису при зверненні до бази даних.

Дані базуються на моделі «сутність–зв’язок». Центральним елементом моделі визначено об’єкт замовлення, який об’єднує інформаційні потоки з суміжних функціональних блоків. Для запобігання виникненню аномалій модифікації, видалення чи оновлення записів на рівні сховища використано принципи типізації та логічної нормалізації схем колекцій.

Таблиця 2.5 - Характеристика інформаційних потоків модуля

Категорія даних	Джерело інформації	Призначення та зміст даних
Вхідні параметри	Клієнт	Сукупність вибору моделі, розміру прикраси та контактних відомостей замовника
Нормативно-довідкові	База даних	Актуальна ціна за грам металу, коефіцієнти складності та доступні залишки
Вихідні показники	Модуль обробки даних	Сформований запис про транзакцію, електронна квитанція та статус виконання

Формування інформаційного забезпечення створює необхідні умови для коректного виконання обчислювальних алгоритмів системи. Завдяки детермінованій класифікації даних та фіксації логічних зв’язків між сутностями, модуль досягає високої точності при розрахунках підсумкової вартості кошика та забезпечує оперативне оновлення інформації про стани поточних сесій, що підвищує загальну продуктивність розробленого програмного продукту.

Процес побудови інформаційного забезпечення модуля управління замовленнями базується на розмежуванні потоків даних. Вхідна інформація – це сукупність даних, які надходять до інформаційної системи із зовнішніх джерел для

подальшого аналізу та перетворення [10]. У межах модуля ці дані є первинним ресурсом, що запускає механізм обслуговування.

Згідно з методологією проектування, інформація має бути достовірною та актуальною. Достовірність інформації - це властивість даних об'єктивно відображати реальний стан об'єктів предметної області [11]. Для забезпечення цієї властивості вхідна інформація підлягає обов'язковій процедурі валідації. Вхідна інформація модуля представлена такими компонентами:

1. Параметричні дані виробу – це вибір специфічних характеристик прикраси, що здійснюється користувачем через веб-інтерфейс. Ці дані ініціюють алгоритми розрахунку та визначають зміст майбутнього замовлення.

2. Персональні реквізити замовника – це відомості, що дозволяють ідентифікувати суб'єкт взаємодії, зокрема прізвище, ім'я, контактний номер телефону та адресу електронної пошти. Ці дані є необхідними для завершення процесу реєстрації замовлення в загальному масиві бази даних.

3. Довідкові дані – це зовнішня інформація про актуальні ціни на дорогоцінні метали та наявність товарних позицій, що надходить із внутрішніх каталогів системи.

Процес трансформації вхідних даних завершується формуванням вихідної інформації. Вихідна інформація – це сукупність результатів функціонування системи, поданих у вигляді, придатному для використання споживачем або для подальшої обробки іншими системами [10]. Вона є логічним підсумком роботи всіх функціональних блоків модуля.

Структура інформаційних одиниць, що обробляються в межах функцій модуля, може бути представлена наступним переліком:

1. Масив характеристик вибору – це фундамент для формування технічного завдання, забезпечуючи точність передачі вимог клієнта.

2. Транзакційні дані – проміжна інформація, що фіксує стан вибору користувача протягом сесії та дозволяє зберігати цілісність обраних товарів до моменту завершення процедури оформлення.

3. Електронна картка замовлення – основний вихідний об’єкт, що містить повний перелік реквізитів, який передається в базу даних для подальшої обробки менеджером.

4. Інформаційне підтвердження – вихідний сигнал модуля, який сповіщає користувача про успішну реєстрацію його запиту та завершення процесу.

Для надійності обміну даними система використовує механізми контролю цілісності. Цілісність інформації - це стан даних, коли вони зберігають свою структуру під час передачі або обробки, не зазнаючи змін. Це важливо для розроблюваного модуля, де найменша помилка у вазі чи пробі виробу може спричинити значні фінансові розбіжності.

Таким чином, опис вхідної та вихідної інформації демонструє цикл даних у системі. Організація потоків дозволяє мінімізувати ймовірність виникнення помилок та забезпечує стабільну роботу модуля, що підвищує ефективність загальної автоматизованої системи управління діяльністю ювелірного інтернет-магазину.

Концептуальне інфологічне проєктування – це етап розробки бази даних, на якому створюється логічна модель предметної області. Вона описує, які саме дані потрібні для роботи модуля та як вони пов’язані між собою, незалежно від обраної системи керування базами даних. Інфологічна модель – це людино-орієнтована модель, яка відображає список даних, необхідних для автоматизації функцій модуля, та фіксує зв’язки між реальними об’єктами предметної області [3, 6].

Головна мета цього етапу – чітко визначити структуру інформації та позбутися дублювання. Процес створення архітектури передбачає поділ загальної задачі на окремі частини, які потім об’єднуються в єдину базу.

Локальні моделі відображають специфічні інформаційні потреби окремих функціональних завдань або груп користувачів, забезпечуючи глибоку деталізацію конкретних операцій. Локальна модель є проєкцією загальної бази даних на потреби конкретного модуля [3, 6].

Для створення модуля виділено три локальні представлення:

1. Модель каталогу товарів: описує характеристики ювелірних виробів, які впливають на відображення асортименту та розрахунок вартості.

2. Модель кошика: відповідає за тимчасове збереження обраних товарів протягом перебування користувача на сайті.

3. Модель оформлення покупки: містить контактні дані клієнта, які необхідні для зворотного зв'язку та підтвердження замовлення.

Пропонована база даних модуля виконання онлайн-замовлень ювелірних прикрас матиме структуру, подану на рисунку 2.5.

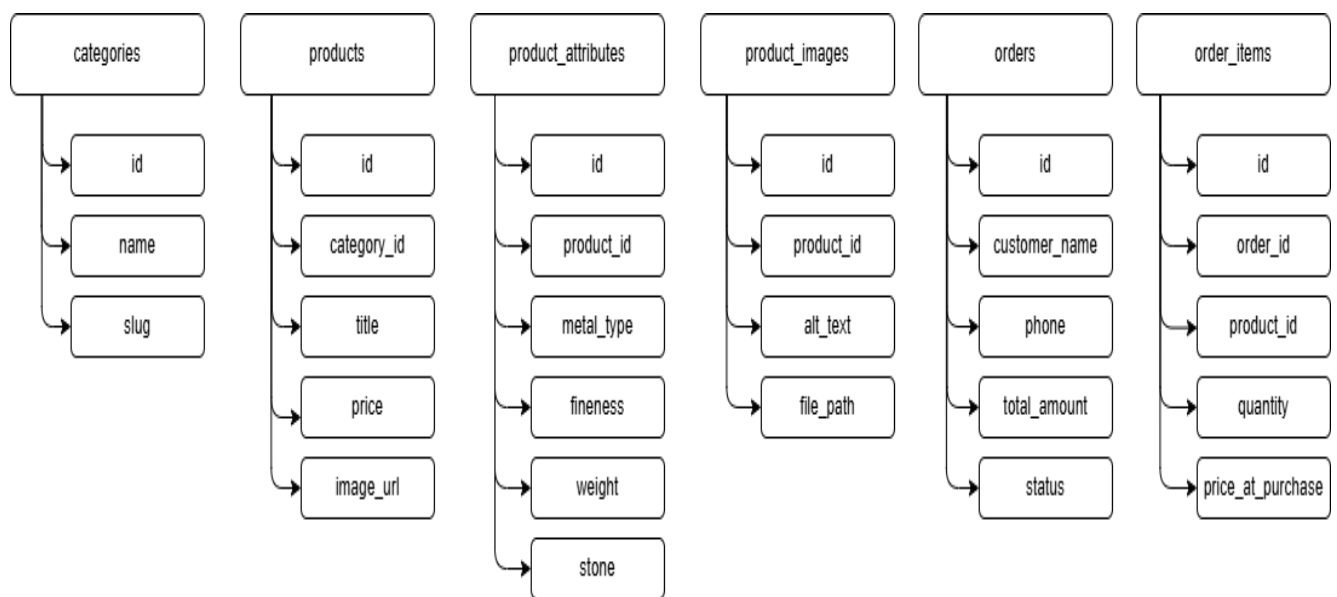


Рисунок 2.5 – Структура таблиць бази даних модуля

Пропонована база даних модуля буде містити таблиці категорій, виробів та їх атрибутів і зображень, замовлень та їх елементів.

Глобальна інфологічна модель створюється шляхом об'єднання цих локальних даних у єдину структуру. Вона базується на методології «сутність–зв'язок» (ER-модель) [12]. Основними складовими цієї моделі є:

1. Сутність «Клієнт»: містить базову контактну інформацію про покупця.
2. Сутність «Товар»: описує ювелірні вироби та зберігає їхні властивості.
3. Сутність «Замовлення»: об'єднує дані клієнта з обраними товарами та зберігає підсумкову суму покупки.

Завершення процесу інфологічного проектування дозволяє досягти високої інформаційної узгодженості системи. Завдяки такому підходу, дані в системі не суперечать одне одному. Це створює надійну основу для подальшого проектування та фізичного створення бази даних.

Даталогічна модель [3, 6, 12] описує структуру колекцій та документів, а не класичних таблиць і реляційних зв'язків між ними. Інформаційний простір модуля базується на двох основних колекціях:

1. Колекція товарів (Products): зберігає асортимент прикрас. Кожен документ містить унікальний ідентифікатор виробу, назву, опис, ціну та перелік доступних розмірів.

2. Колекція замовлень (Orders): фіксує результати покупок. Вона зберігає дату транзакції, підсумкову суму, контактні дані покупця та вкладений масив обраних товарів.

Для забезпечення ефективного опрацювання даних, спроектовано діаграму класів, що відображає архітектуру основ бізнес-логіки. UML-діаграма (рисунок 2.6) відображає зв'язки між об'єктами Product, Cart та Order, які утворюють фундамент функціоналу інтернет-магазину.

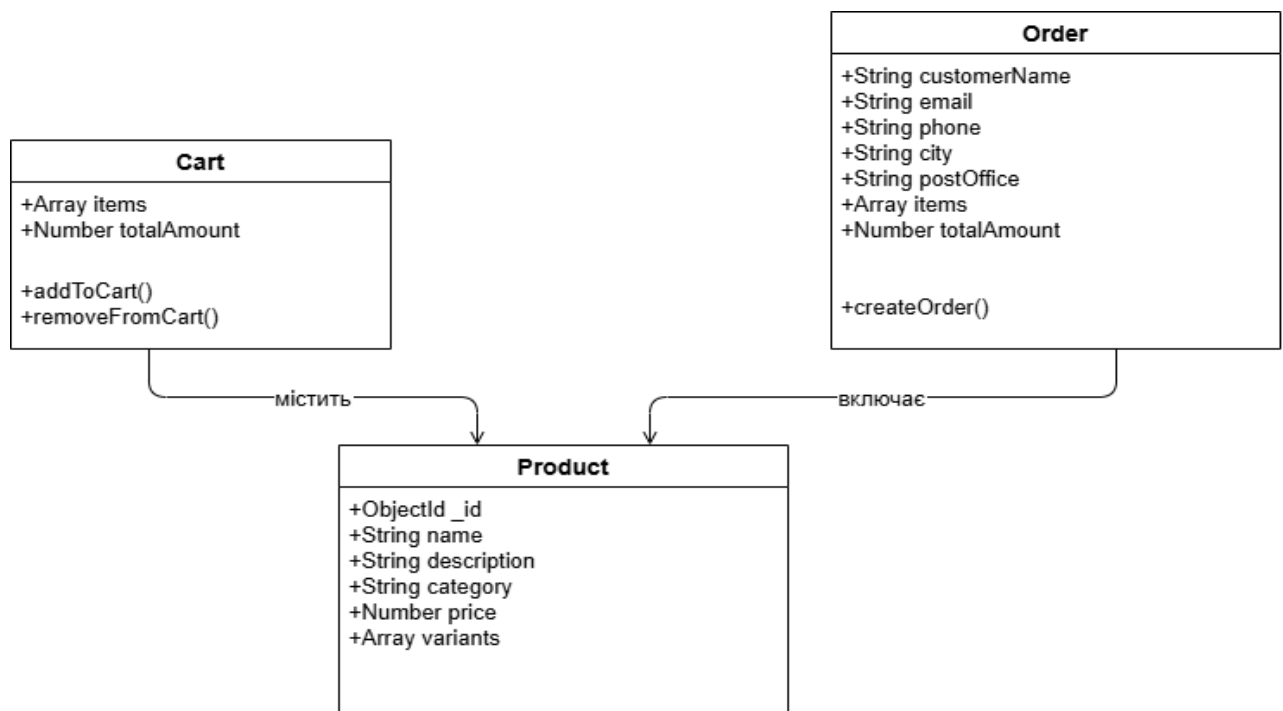


Рисунок 2.6 –Діаграма класів

Клас Cart, відповідає за тимчасове збереження обраних позицій під час сесії користувача. Він містить масив елементів та динамічно вираховує загальну вартість покупок (таблиця 2.6). Логіка цього класу забезпечується двома основними функціями, які дозволяють додавати нові товари до кошика або видаляти їх у разі потреби. На діаграмі цей зв'язок підписано словом «містить», що вказує на безпосереднє наповнення кошика об'єктами товарів.

Таблиця 2.6 – Компоненти та типи даних класу Cart

Тип компонента	Назва	Тип даних	Опис та призначення
Атрибут	+Array items	Масив	Масив товарів, які додані до кошика
Атрибут	+Number totalAmount	Число	Загальна вартість товарів у кошику
Метод	+addToCart()	Функція	Функція для додавання товару до кошика
Метод	+removeFromCart()	Функція	Функція для видалення товару з кошика

Клас Product – описує картку товару в каталозі магазину. Кожен товар має системний ідентифікатор, назву, опис, категорію та вартість (таблиця 2.7). Структура передбачає масив варіантів, що дозволяє зберігати модифікації одного продукту, наприклад, розбіжності за розмірами чи матеріалами.

Клас Order – відповідає за фінальний етап покупки та фіксує інформацію про оформлене замовлення клієнта (таблиця 2.8). Окрім аналогічного масиву замовлених товарів та підсумкової суми, цей клас акумулює всі необхідні персональні й логістичні дані покупця. До них належать ім'я та прізвище клієнта, його електронна пошта, контактний номер телефону, а також місто та поштове відділення для доставки. Завершальним процесом у цьому ланцюжку є метод створення замовлення, який реєструє всі зібрані дані в базі. Спрямована до товару стрілка з підписом «включає» демонструє, що кожне замовлення обов'язково складається з переліку товарів, придбаних користувачем.

Таблиця 2.7 – Компоненти та типи даних класу Product

Тип компонента	Назва	Тип даних	Опис та призначення
Атрибут	+ObjectId _id	ObjectId	Системний ключ документа, який MongoDB автоматично створює для кожного товару
Атрибут	+String name	String	Рядок із назвою виробу для відображення в інтерфейсі картки товару
Атрибут	+String description	String	Рядок із описом характеристик, дизайну чи властивостей прикраси.
Атрибут	+String category	String	Рядок-тег категорії, для сортування та навігації
Атрибут	+Number price	Number	Числове значення вартості однієї одиниці виробу
Атрибут	+Array variants	Array	Масив модифікацій прикраси
Метод	Методи відсутні	–	У початковому коді цей клас діє виключно як модель/схема даних, тому секція методів залишається порожньою

Таблиця 2.8 – Компоненти та типи даних класу Order

Тип компонента	Назва	Тип даних	Опис та призначення
Атрибут	+String customerName	String	Ім'я та прізвище клієнта для доставки
Атрибут	+String email	String	Електронна адреса для зв'язку та інформування
Атрибут	+String phone	String	Контактний номер телефону покупця
Атрибут	+String city	String	Місто або населений пункт для доставки
Атрибут	+String postOffice	String	Відділення пошти
Атрибут	+Array items	Array	Масив товарів, перенесений із кошика в бланк замовлення
Атрибут	+Number totalAmount	Number	Підсумкова сума замовлення до сплати, що підлягає збереженню
Метод	+createOrder()	Function	Функція, яка збирає дані форми, формує фінальний об'єкт і викликає процедуру запису замовлення в базу даних

Для створення логіки взаємодії клієнта з платформою та детального аналізу поведінки графічних елементів була застосована UML-діаграма станів (рисунок 2.7). Цей інструмент дозволяє відобразити переходи компонентів фронтенд-частини з одного стабільного режиму в інший під дією тригерів, спричинених активністю користувача.



Рисунок 2.7 – Діаграма станів

Динаміка інтерфейсу побудована на основі переходу між ключовими етапами вибору товарів та оформлення покупки [17]:

1. Стан «Каталог»: Є вихідним станом інтерфейсу. Усі елементи навігації та картки товарів перебувають у режимі очікування дій клієнта.

2. Стан «Перегляд Товару»: фокусує увагу користувача на детальному описі ювелірного виробу. У цьому стані інтерфейс дозволяє переглянути деталі або ініціювати конфігурацію параметрів.

3. Стан «Конфігурація»: Активує елементи керування для вибору параметрів прикраси. Кнопка "Додати до кошика" стає активною лише після фіксації параметрів.

4. Стан «Активний Кошик»: Відображає динамічний вміст обраних товарів. UI виконує миттєвий перерахунок загальної вартості (totalAmount) при зміні кількості одиниць.

5. Стан «Оформлення Замовлення»: Переводить інтерфейс у режим збору персональних. Поля введення перебувають у стані очікування тексту.

6. Стан «Валідація Даних»: Логічний вузол, який на рівні інтерфейсу перевіряє коректність заповнення форми. Залежно від результату, UI переходить в один із двох станів. «Помилка Введення»: Інтерфейс візуально підсвічує некоректні поля та виводить валідаційні повідомлення. «Обробка Замовлення (UI-завантаження)»: Інтерфейс у стані очікування відповіді від сервера, блокуючи повторне надсилання форми та відображаючи індикатор завантаження.

7. Стан «Підтвердження Успіху (UI)»: Фінальний стан, який візуалізує успішне завершення транзакції та ініціює очищення стану локального кошика.

3 ПРОГРАМНО-ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Реалізація бази даних

Програмна реалізація – це процес створення фізичних структур даних, встановлення правил цілісності та налаштування механізмів доступу до інформації за допомогою інструментів конкретної СКБД. Для програмної реалізації було обрано документоорієнтовану СКБД MongoDB [21]. Вибір зумовлений гнучкістю схеми даних, що дозволяє легко масштабувати структуру товарів без складних SQL-міграцій, та високою швидкістю роботи з JSON-об'єктами, які ідентичні формату передачі даних у стейку MERN.

Програмну реалізацію модуля виконано на основі стека технологій MERN (MongoDB, Express, React, Node.js) [21-24], що забезпечує повний цикл обробки даних – від клієнтського інтерфейсу до сервера та бази даних.

Програмна логіка сервера побудована за допомогою платформи Node.js та фреймворка Express. Для збереження даних та забезпечення структури карток товарів обрано NoSQL базу даних MongoDB. У процесі роботи використано принципи об'єктно-орієнтованого програмування та архітектурний патерн REST API. Моделювання структури бази даних та зв'язків між модулями виконано за допомогою мови UML. Написання коду здійснювалося в середовищі Visual Studio Code, а верифікація серверних запитів – за допомогою інструменту Postman.

Інтерфейс розроблено на базі бібліотеки React, що забезпечує високу швидкість роботи сторінок, компонентний підхід та плавний користувацький досвід у мінімалістичному стилі.

Цілісність даних контролюється на рівні схем Mongoose. Кожен документ автоматично отримує унікальний ідентифікатор, а обов'язкові поля проходять валідацію, що виключає запис неповних або помилкових даних.

Архітектура системи побудована за принципом клієнт-серверної взаємодії, де кожен рівень має чітко визначену функціональність. Клієнтський модуль (рисунок 3.1) – відповідає за відображення контенту та обробку взаємодії користувача. Включає компоненти для каталогу, кошика та сторінки бренду.

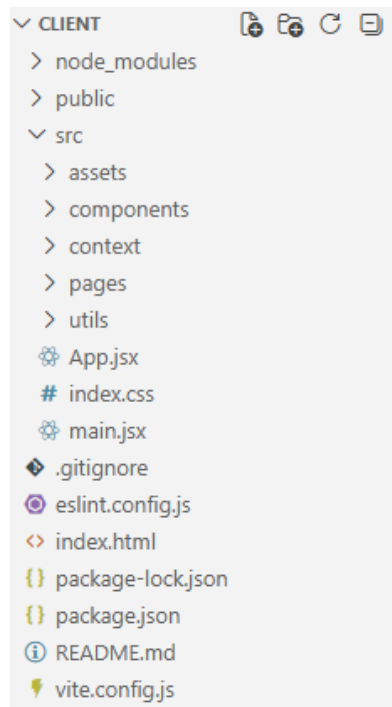


Рисунок 3.1 - Frontend (папка client)

Серверний модуль (рисунок 3.2) – це посередник між клієнтом та базою даних. Забезпечує автентифікацію, валідацію замовлень та обробку запитів [5].

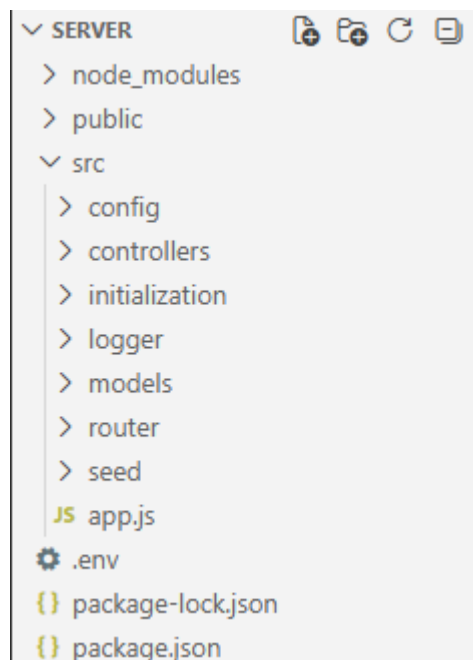


Рисунок 3.2 - Backend (папка server)

База даних [2, 20, 21] забезпечує збереження інформації про товари, а також клієнтів, використовуючи документоорієнтовану модель.

Фізичне проєктування бази даних є фінальним етапом розробки інформаційного забезпечення, на якому розроблена даталогічна структура адаптується до специфіки обраного програмного середовища. Як зазначено вище, для реалізації бази даних у межах даного проєкту обрано MongoDB (рисунок 3.3) - це база даних, що належить до NoSQL систем та використовує гнучку схему документів у форматі BSON [17]. Вибір даної технології обумовлений потребою у високій швидкості обробки запитів та гнучкості структури даних при описі складних ювелірних виробів.

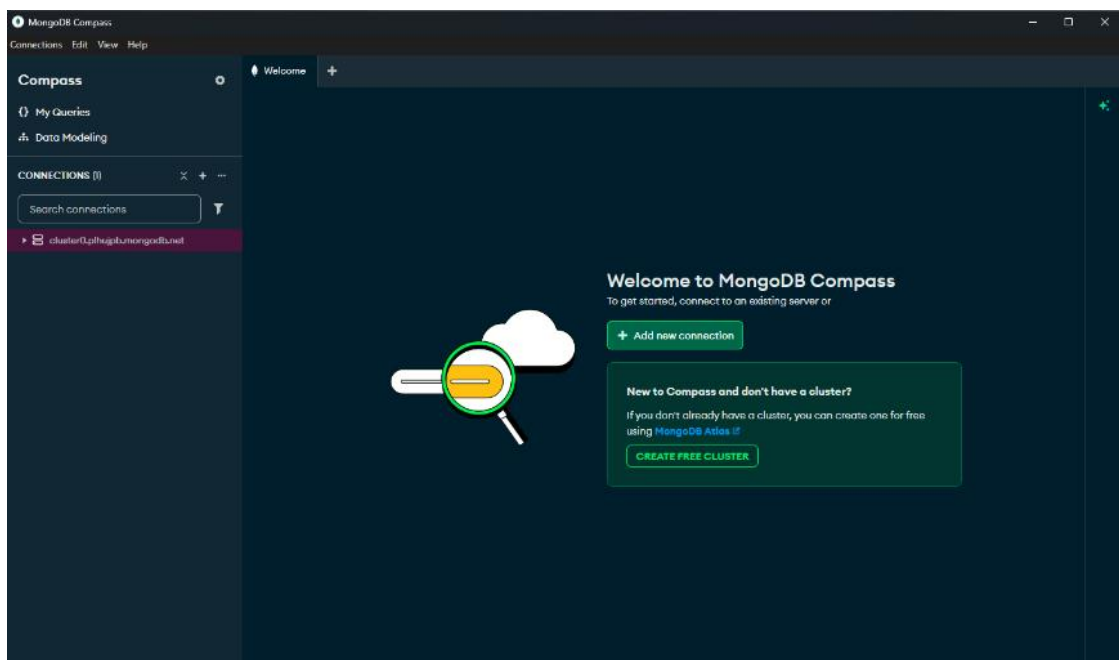


Рисунок 3.3 – Інтерфейс MongoDB Compass

Фізична організація даних у проєкті на базі MongoDB має особливості:

1. На відміну від класичних реляційних таблиць, інформація в модулі організована у вигляді колекцій документів. Це дозволяє використовувати принцип вкладеності (embedding), де параметри вибору клієнта (тип металу, проба, розмір) зберігаються безпосередньо у документі замовлення, що значно прискорює завершення процесу зчитування даних.

2. MongoDB характеризується відсутністю жорстко визначеної схеми (schemaless), що дозволяє легко додавати нові атрибути виробів без необхідності перебудови всієї бази даних [21].

3. Для оптимізації доступу до інформації у фізичній моделі передбачено створення індексів за ключовими полями, такими як контактний телефон клієнта. Це гарантує мінімальний час відгуку системи навіть при значному обсязі накопичених записів.

Такий підхід забезпечує стабільність роботи модуля під навантаженням та спрощує інтеграцію бази даних із веб-інтерфейсом [2]. Завершення етапу фізичного проектування створює готовий технічний фундамент для розгортання системи та переходу до безпосередньої програмної реалізації алгоритмів.

У ході реалізації бази даних для модуля виконання замовлень в ювелірному інтернет-магазині виконуються такі програмні кроки:

1. Створено колекції (рисунок 3.4), які є контейнерами для документів у форматі BSON.

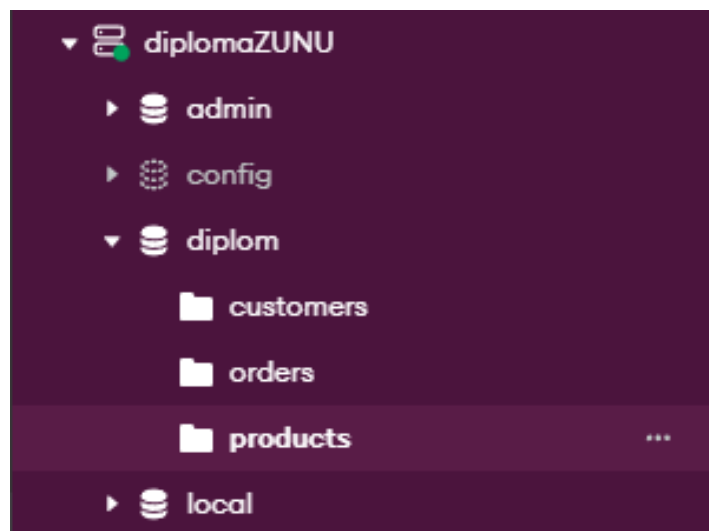


Рисунок 3.4 – Створені колекції у MongoDB Compass

2. Кожна прикраса записується як окремий документ, що містить об'єкти з параметрами виробу (рисунок 3.5), що дозволяє уникнути складних операцій об'єднання при зчитуванні даних.

```

1 {
2   "_id": "6a061918cf7126c82d8a41ed",
3   "name": "Каблучка 'Зоряна ніч'",
4   "description": "Елегантна прикраса з фіанітами",
5   "category": "Каблучки",
6   "variants": [
7     {
8       "metalName": "Золото",
9       "colorCode": "#FFD700",
10      "price": 12500,
11      "imageUrl": "/images/gold_ring.jpg",
12      "sizes": [
13        /* Array (2) */
14      ],
15      "_id": "6a061918cf7126c82d8a41ee"
16    },
17    {
18      "metalName": "Срібло",
19      "colorCode": "#C0C0C0",
20      "price": 1800,
21      "imageUrl": "/images/silver_ring.jpg",
22      "sizes": [
23        /* Array (2) */
24      ],
25      "_id": "6a061918cf7126c82d8a41f1"
26    }
27  ],
28   "__v": 0
29 }

```

Рисунок 3.5 – Об’єкти з параметрами виробу

3. Встановлюється з’єднання між програмним кодом модуля та сервером бази даних за допомогою драйвера, що забезпечує стабільну передачу інформації (рисунок 3.6).

```

1 MONGODB_URL=mongodb+srv://leria:qwerty123@cluster0.plhujpb.mongodb.net/diplom?retryWrites=true&w=
  =majority&appName=Cluster0
2 CLIENT_URL=http://localhost:5173
3 SERVER_URL=http://localhost:8080
4 SERVER_PORT=8080
5 NODE_ENV=development

```

Рисунок 3.6 - Фрагмент підключення модуля до сервера бази даних.

Інформаційні зв’язки реалізовані через RESTful API. Взаємодія відбувається шляхом передачі HTTP-запитів, де клієнт виконує операції: створення, читання, оновлення, або ж видалення, а сервер виконує відповідні маніпуляції з колекціями БД, повертаючи відповіді у форматі JSON.

Для побудови інтерфейсу платформи використано бібліотеку React [19, 22], розділивши застосунок на логічні сторінки та модулі: каталог товарів, кошик, форму замовлення та інформаційний блок «Про нас». Маршрутизація між цими

розділами відбувається динамічно на стороні клієнта. Життєвий цикл усього інтерфейсу та логіку його станів контролює інструмент React Context API, а також базові хуки `useState` та `useEffect`.

Усі методи, класи та функції керування інтерфейсом описуються спеціальними документаційними тегами специфікації JSDoc. Це сучасний галузевий стандарт коментування для JavaScript.

Детальну структуру схем Mongoose, що визначають логіку зберігання даних, наведено у додатку А (лістинг А.1 та А.2). Повний програмний код, який відповідає за роботу контексту кошика, зміну характеристик ювелірних виробів на сторінках каталогу та перевірку екранних форм, разом із JSDoc-коментарями подано у додатку Б.

Під час проєктування платформи було обрано гнучкий та сучасний технологічний стек MERN (MongoDB, Express.js, React, Node.js). Головний фактор цього вибору – заміна різних мов програмування на єдиний стандарт JavaScript/TypeScript [17] на всіх рівнях системи. Такий крок спрощує та пришвидшує обмін даними між користувачем та сервером.

Бібліотеку React [19, 22] – задіяно через її підхід та високу швидкість оновлення інтерфейсу завдяки технології Virtual DOM. Це дозволяє забезпечити миттєвий перехід між каталогом товарів, інформаційними розділами та кошиком без перезавантаження всього сайту в браузері. Інтерфейс розподілено на окремі блоки. Зокрема, селектори для вибору матеріалу чи розміру прикраси реалізовано як автономні компоненти, що робить можливим повторне використання в різних частинах проєкту.

Для стабільної роботи системи, за умови сильного навантаження було обрано середовище Node.js [18, 24]. Для реалізації логіки маршрутизації було використано фреймворк Express.js [18, 23]. Також цей інструмент спрощує підключення захисних засобів для модулів обробки інформації про замовлення.

У NoSQL-системі MongoDB [21] зберігання інформації організовано у вигляді гнучких документів, схожих на JSON-файли. Ювелірна продукція має

багато характеристик, тому гнучка структура MongoDB дозволяє миттєво додавати нові властивості для товарів без необхідності перезапуску всієї бази даних.

3.2 Інтерфейс користувача

Проектування ефективного графічного середовища потребує детального аналізу потенційних сценаріїв поведінки відвідувачів, чіткого розуміння їхніх практичних завдань, а також виявлення специфічних вимог до зручності використання програмного продукту. Під час розробки візуальної частини платформи враховано та інтегровано принципи побудови сучасних цифрових інтерфейсів (рисунок 3.7).

Інтерфейс допомагає користувачеві оперативно вирішувати його поточні завдання під час взаємодії з платформою. Програму зорієнтовано на максимальну допомогу під час взаємодії, що запобігає виникненню штучних перешкод чи технічних бар'єрів.

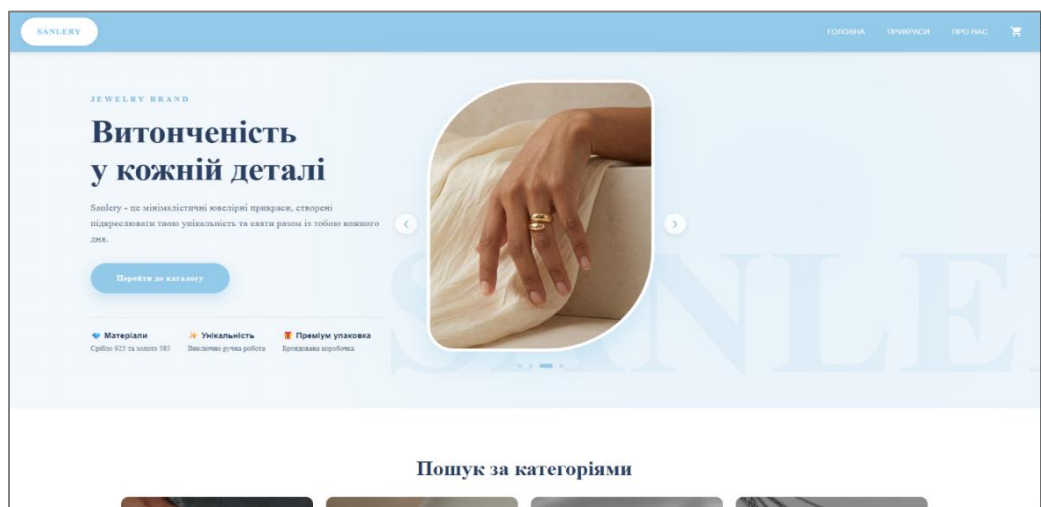


Рисунок 3.7 –Головне вікно

Успішне виконання цього завдання забезпечено впровадженням інженерних та дизайнерських підходів. Завдяки інтеграції лаконічного навігаційного меню відкрито швидкий і зрозумілий доступ до головних сторінок, включаючи каталог, інформаційний блок із описом бренду та форму підтвердження замовлення.

Для керування станом кошика та динамічного оновлення вартості обраних виробів застосовано механізм контексту React. Логіку додавання товарів, перерахунку загальної суми та валідації вибору характеристик реалізовано в компоненті CartContext.js. Повний текст програмного коду цієї логіки подано у додатку Б (лістинг Б.1).

Натомість розробку зорієнтовано на принципи вільного простору, суворой візуальної підпорядкованості компонентів та виразної контрастної типографіці. Такий підхід допомагає користувачеві не відволікатися на другорядні деталі й фокусувати всю увагу лише на вивченні специфікацій продукції.

Для детального ознайомлення користувача з інформацією про філософію бренду та технологічні стандарти розроблено окремий інформаційний модуль (рисунок 3.8).



Рисунок 3.8 – Інтерфейс інформаційного модуля «Про нас»

Під час роботи з вебплатформою враховані фактори, які можуть викликати у користувача відчуття дискомфорту, розгубленості або невпевненості в коректності своїх дій. Організацію середовища взаємодії забезпечено впровадженням наступних інженерних правил:

1. У системі впроваджено інтелектуальні алгоритми верифікації вхідних даних (рисунок 3.9). Перевірку результатів активовано для важливих операцій.

Рисунок 3.9 – Інтерфейс в режимі обробки помилок валідації даних

2. Графічну систему спроектовано, щоб чітко вказувати користувачеві на необхідну дію. Інформаційні повідомлення та діалогові вікна з'являються виключно в моменти реальної потреби – наприклад, під час успішного додавання прикраси у кошик (рисунок 3.10).

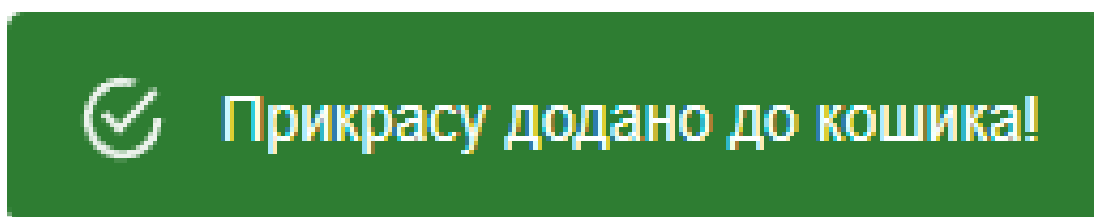


Рисунок 3.10 - Інтерфейсне вікно успішного додавання
прикраси у кошик

3. Зміст усіх текстових повідомлень, які виводяться на екран, ретельно продумано та відредаговано. Застосовано зрозумілі формулювання, які констатують факт виникнення проблеми та пропонують чіткий алгоритм для її швидкого вирішення.

Використання зазначених принципів дозволило створити збалансований, передбачуваний та зручний користувацький інтерфейс, який забезпечує високий рівень ефективності та позитивний клієнтський досвід (User Experience).

3.3 Тестування модуля

З метою підтвердження працездатності, стабільності та відповідності створеного програмного продукту встановленим технічним вимогам проведено комплексне тестування системи. Для перевірки задіяно метод «чорної скриньки» (Black Box Testing), що дозволяє оцінити коректність функціонування інтерфейсу та логіки без аналізу внутрішньої структури вихідного коду. Для систематизації процесу перевірки реалізовано кілька базових тестових сценаріїв (таблиця 3.1).

Таблиця 3.1 – Протокол тестування функціональних модулів

№	Назва та мета сценарію	Вхідні дії (кроки)	Очікуваний результат	Фактичний результат	Статус
1	Перевірка фільтрації в каталозі	Вибір категорії прикрас та фільтра.	Відображення товарів, які відповідають критеріям.	Перелік товарів оновлено згідно з фільтром.	Успішно
2	Додавання товару з унікальними параметрами	Обрання розміру, типу металу та клік «Додати до кошика».	Товар з'являється у кошику із зафіксованими параметрами.	Позицію успішно додано до кошика, стан оновлено.	Успішно
3	Об'єднання дублікатів у кошику	Повторне додавання аналогічного товару з тим самим розміром і металом.	Новий рядок не створюється, кількість збільшується на 1.	Кількість товару, загальну суму перераховано.	Успішно
4	Валідація форми при порожніх полях	Натискання кнопки підтвердження замовлення без введення імені та телефону.	Блокування відправки запиту, підсвічування обов'язкових полів.	Клієнтська валідація спрацювала, виведено повідомлення про помилки.	Успішно
5	Успішне оформлення покупки	Заповнення всіх полів коректними даними та відправка форми.	Надсилання даних на сервер, очищення кошика, екран успіху.	Запит оброблено, локальний кошик анульовано, виведено підтвердження.	Успішно

Основними завданнями тестування визначено перевірку навігації, коректності обробки стану кошика при виборі різних модифікацій виробів, а також валідацію форм збору даних.

Аналіз каталогу (рисунок 3.11) підтверджує спроможність системи оперативно реагувати на встановлені користувачем критерії сортування та динамічно звужувати перелік доступних для огляду ювелірних прикрас. Під час активації елементів фільтрації клієнтська частина застосунку миттєво перераховує масив даних.

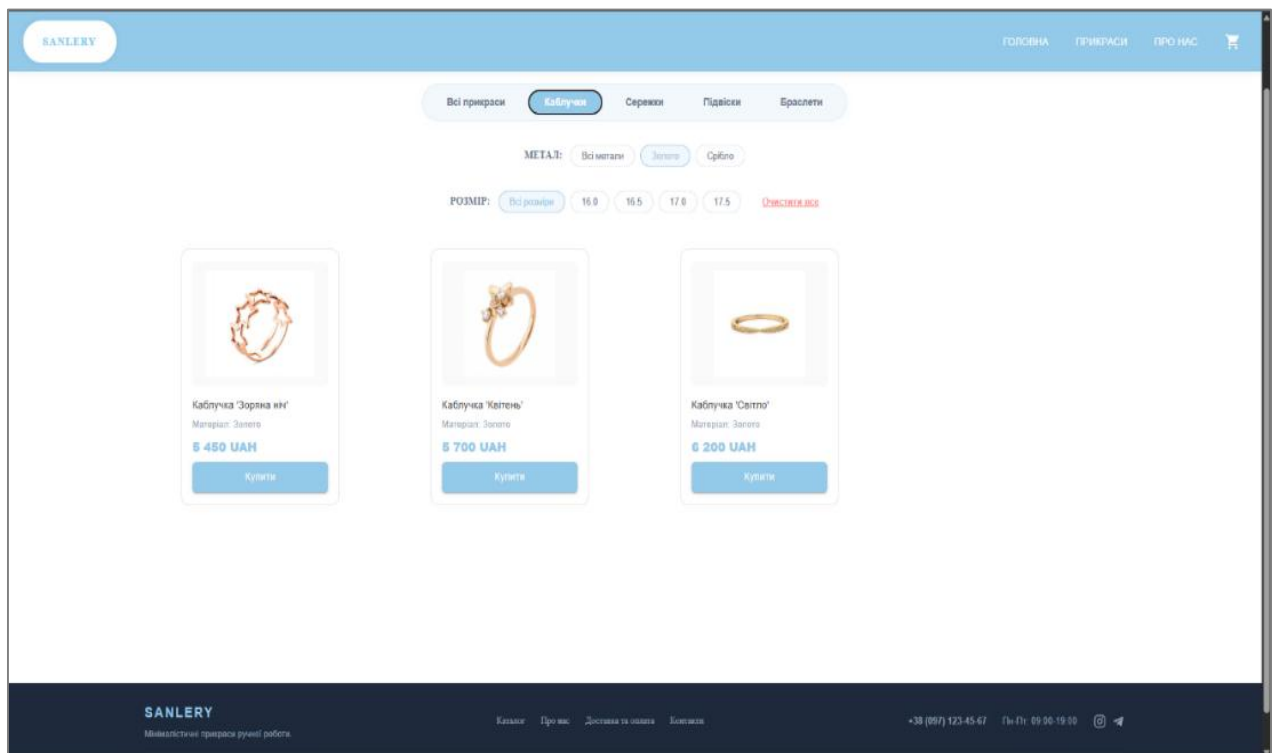


Рисунок 3.11 - Графічний результат тестування модуля фільтрації товарів у каталозі

Коректність роботи програмного контексту кошика, який здатний диференціювати вироби не лише за базовим ідентифікатором, а й за сукупністю обраних фізичних модифікацій (рисунок 3.12).

Фінальну стадію інтеграційного тестування, що фіксує перехід інтерфейсу (рисунок 3.13) у кінцевий стан після успішного підтвердження транзакції сервером.

Додатково було проведено інтеграційне тестування серверної частини, що включало перевірку коректності відповідей API-контролерів при зверненні до бази даних MongoDB. Тестування продемонструвало стабільну обробку запитів GET та

POST, а також правильну роботу валідації даних на рівні схем Mongoose (код контролерів та роутерів наведено у додатку Б, лістинги Б.2, Б.3).

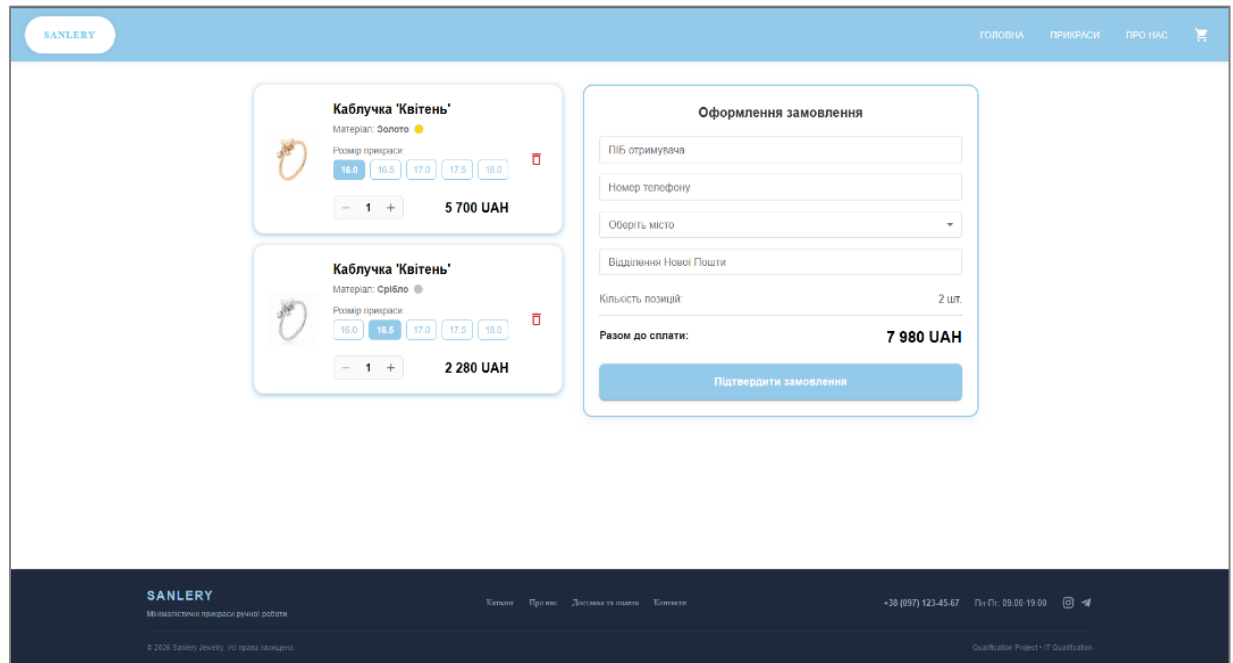


Рисунок 3.12 - Перевірка коректності розподілу та відображення модифікацій у кошику

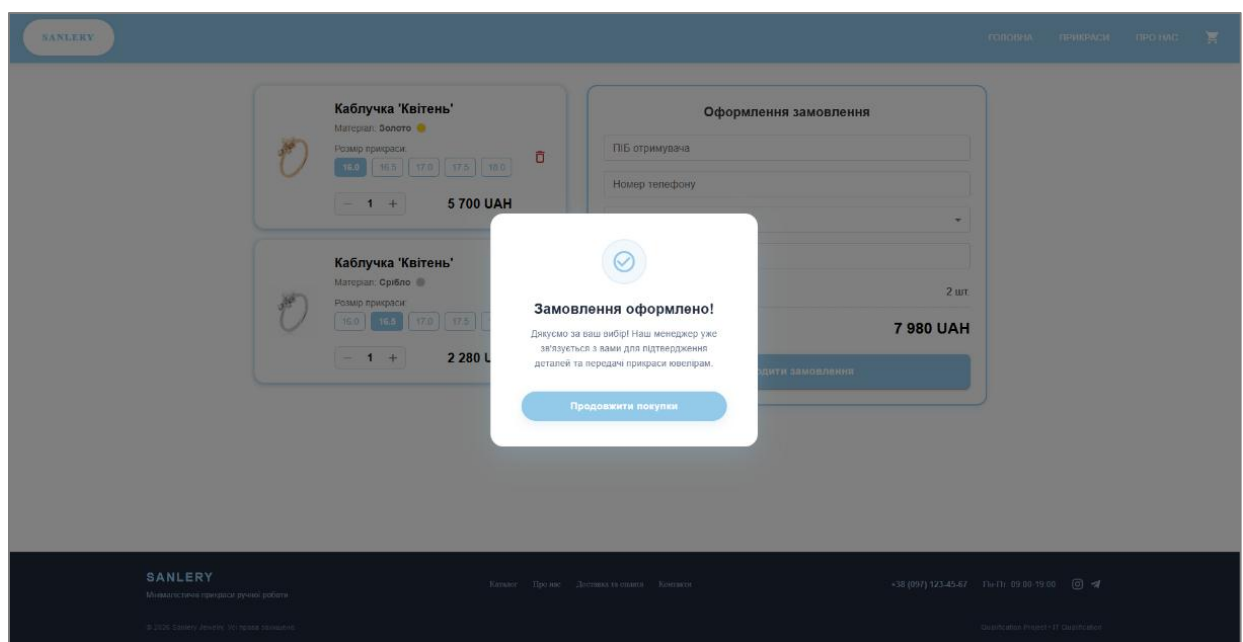


Рисунок 3.13 - Візуалізація користувацького інтерфейсу при успішній фіксації замовлення

На основі аналізу проведених операцій та виконання контрольних тестових сценаріїв встановлено, що всі зафіксовані тести реалізуються без критичних помилок або функціональних збоїв. Клієнтська частина веб-платформи демонструє передбачувану та стабільну обробку станів та інтегрованої бізнес-логіки.

Реактивна поведінка компонентів, асинхронна взаємодія під час ініціалізації запитів до серверної частини, а також механізми збереження інформації у локальному сховищі браузера повністю відповідають встановленим критеріям надійності, що гарантує комфортну та безперебійну роботу користувача з ресурсом.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи:

1. Проаналізовано особливості інтернет-ринку ювелірних виробів. Встановлено, що в сучасних умовах актуальним є створення програмного продукту для онлайн-замовлень ювелірних виробів.

2. На основі порівняльного аналізу існуючих ринкових аналогів обґрунтовано доцільність розробки програмного модуля для виконання онлайн-замовлень ювелірних прикрас.

3. Сформульовано вимоги до програмного модуля та завдання для досягнення мети його створення.

4. Спроектовано архітектуру бази даних та логіку взаємодії клієнтської і серверної частин модуля. Гнучка структура збереження даних забезпечує легке масштабування платформи при майбутньому розширенні асортименту ювелірних виробів.

5. Розроблено адаптивні компоненти користувацького інтерфейсу в лаконічному візуальному стилі.

6. Програмно реалізовано серверні сервіси для обробки операцій та керування контентом.

7. Проведено тестування розробленого модуля. Результати вказують на коректність роботи та достатню швидкодію. Перевірка модуля методами функціонального тестування підтвердила стабільність бізнес-логіки та точність валідації даних.

8. Практична цінність отриманих результатів полягає у створенні готового до експлуатації програмного продукту, спрямованого на виконання онлайн-замовлень ювелірних виробів. У якості перспективних напрямків подальшого вдосконалення модуля планується автоматизація сповіщень клієнтів про нові поступлення товарів та впровадження алгоритмів персональних рекомендацій.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Береза А. М. Основи проєктування інформаційних систем: навч. посіб. К.: КНЕУ, 2005. 234 с.
2. Бублик С. Г. Бази даних: підручник. К.: Видавнича група BVH, 2011. 248с.
3. Гринченко М. М. Проєктування інформаційних систем: навч. посіб. К.: КНУ, 2019. 180 с.
4. Дейт К. Дж. Вступ до систем баз даних: пер. з англ. К.: Діалектика, 2016. 784 с.
5. Карпенко Ю. О. Веб-технології та розробка програмного забезпечення. Харків: ХНУРЕ, 2022. 210 с.
6. Ковтун В. В. Алгоритмізація та програмування: навч. посіб. К.: Освіта, 2020. 250 с.
7. Литвин В. В. Технології розробки веб-додатків: навч. посіб. Львів: Львівська політехніка, 2022. 210 с.
8. Мельник А. О. Інформаційні системи: підручник. Львів: БаК, 2019. 450 с.
9. Носовець О. К. Проєктування інформаційних систем: теорія та практика. Київ: КНУБА, 2020. 190 с.
10. Пасічник В. В., Резніченко В. А. Організація баз даних та знань: підручник. К.: Видавнича група BVH, 2006. 384 с.
11. Петренко А. І. Проєктування інтернет-ресурсів: посібник. К.: Просвіта, 2023. 220 с.
12. Романюк О. Н. Проєктування інтелектуальних систем: навч. посіб. Вінниця: ВНТУ, 2021. 240 с.
13. Ткачук В. М. Проєктування інформаційних систем: навч. посіб. Івано-Франківськ: ІФНТУНГ, 2018. 200 с.
14. Федорова Т. Ю. Розробка веб-додатків на JavaScript. Одеса: ОНПУ, 2024. 150 с.
15. Цимбал В. П. Основи побудови інформаційних систем. Київ: КНЕУ, 2022. 260 с.

16. Юрченко О. А. Веб-дизайн та UX/UI принципи. Харків: ХНЕУ, 2023. 180 с.
17. Фленаган Д. JavaScript: The Definitive Guide. 7th ed. O'Reilly Media, 2020. 704 с.
18. Браун Е. Web Development with Node and Express. 2nd ed. O'Reilly Media, 2019. 352 с.
19. Бенкс А., Порсело Є. Learning React. 2nd ed. O'Reilly Media, 2020. 360 с.
20. Хокінс Т. MongoDB Data Modeling. O'Reilly Media, 2021. 220 с.
21. MongoDB Manual. URL: <https://www.mongodb.com/docs/>
22. React.dev Documentation. URL: <https://react.dev/>
23. Express.js API Reference. URL: <https://expressjs.com/>
24. Node.js Documentation. URL: <https://nodejs.org/docs/>
25. Pandora/Tiffany & Co. Official sites. URL: <https://epandora.ua/> ; <https://www.tiffany.com/>
26. Tiffany & Co. Official Site. URL: <https://www.tiffany.com/> .
27. Tiffany & Co. Customer Reviews on Trustpilot. URL: <https://www.trustpilot.com/review/www.tiffany.com> .
28. Золотий Вік : офіційний сайт. URL: <https://zlotiyvik.ua/ua/>
29. Просування сайту ювелірного магазину: стратегії для зростання продажів URL: <https://seomarket.ua/blog/prosuvannia-sajtu-iuvelirnogo-mahazynu-stratehii-dlia-zrostannia-prodazhiv/>
30. <https://ain.ua/2025/10/21/123-mlrd-grn-jewels/>
31. Котрубенко В., Васильків Н.М. Програмна система онлайн-замовлень ювелірних прикрас. *ICSPM 2026: Інтелектуальні обчислювальні системи та управління проєктами: зб. тез доповідей І міжнародної науково-практичної конференції студентів і молодих вчених (21-22 травня 2026р. Тернопіль).* 2026. С. 184-186.
32. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Лип'яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122

«Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль:
ЗУНУ, 2024. 52 с.

Додаток А
Лістинги моделей

Додаток Б

Програмний код модуля замовлень

Додаток В
Копія публікації