

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему СТВОРЕННЯ ПРОГРАМНОЇ СИСТЕМИ ОПРАЦЮВАННЯ
ДАНИХ ДЛЯ ПРОДАЖУ АНТИКВАРІАТУ
DESIGN OF A DATA PROCESSING SYSTEM FOR
THE SALE OF ANTIQUES

Виконав(ла): студент(ка) 4 курсу, групи КНТ-222

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

ПИСАНЕЦЬ Д.В.

(ПРІЗВИЩЕ та ініціали)

Керівник ОЛІЙНИК А.О.

(ПРІЗВИЩЕ та ініціали)

Рецензент ШИТІКОВА О.В.

(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 122 Комп'ютерні науки
(код і найменування)
Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2026 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ПИСАНЦЯ Дениса Віталійовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Створення програмної системи опрацювання даних для продажу антикваріату. Design of a Data Processing System for the Sale of Antiques

керівник проєкту (роботи) д.т.н., професор, ОЛІЙНИК Андрій Олександрович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 7 ” квітня 2026 року № 139

2. Строк подання студентом проєкту (роботи) 03 червня 2026 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис системи опрацювання даних. 4. Експлуатація, тестування та експериментальне дослідження системи опрацювання даних.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ОЛІЙНИК А.О., професор		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання “ 7 ” квітня 2026 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури системи опрацювання даних.	2 тиждень	Розділ 3
5	Розробка системи опрацювання даних.	3-4 тижні	Розділи 3, 4
6	Тестування та експериментальне дослідження системи опрацювання даних.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

_____ Денис ПИСАНЕЦЬ
(підпис) (Імя ПРІЗВИЩЕ)

Керівник проєкту (роботи)

_____ Андрій ОЛІЙНИК
(підпис) (Імя ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
86 с., 4 табл., 30 рис., 3 дод., 20 джерел.

MYSQL, PHP, VISUAL STUDIO CODE, БАЗА ДАНИХ, МОВА ПРОГРАМУВАННЯ, ПРОДАЖ ТОВАРІВ, СИСТЕМА ОПРАЦЮВАННЯ ДАНИХ.

Об'єкт дослідження – процеси обчислень, пов'язані з організацією та опрацюванням даних для продажу антикваріату.

Предмет дослідження – програмні системи опрацювання даних для продажу антикваріату.

Мета роботи – розробка програмної системи опрацювання даних для продажу антикваріату для підвищення ефективності розповсюдження товарів.

Матеріали, методи та технічні засоби: мова програмування PHP, середовище розробки Visual Studio Code, система управління базами даних MySQL.

Результати. Розроблено проєктні рішення для створення системи опрацювання даних для продажу антикваріату. Створено систему опрацювання даних для продажу антикваріату. Проведено дослідження розробленої системи опрацювання даних для продажу антикваріату.

Висновки. Мету роботи досягнуто. Розроблено систему опрацювання даних для продажу антикваріату за допомогою мови програмування PHP, середовища розробки Visual Studio Code та системи управління базами даних MySQL.

Галузь використання – продаж товарів через Інтернет.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 86 pages, 4 tables, 30 figures, 3 appendixes, 20 sources.

MYSQL, PHP, VISUAL STUDIO CODE, DATABASE, PROGRAMMING LANGUAGE, SALES OF GOODS, DATA PROCESSING SYSTEM.

The object of research is computing processes related to the organization and processing of data for the sale of antiques.

The subject of the research is data processing systems for the sale of antiques.

The purpose of this work is to develop the data processing system for the sale of antiques to increase the efficiency of the distribution of goods.

Materials, methods and technical tools: PHP programming language, Visual Studio Code development environment, MySQL database management system.

Results. Project solutions for the creation of data processing system for the sale of antiques has been designed. The data processing system for the sale of antiques has been developed. The study of the developed data processing system for the sale of antiques has been conducted.

Conclusions. The goal of the work has been achieved. The data processing system for the sale of antiques using the PHP programming language, the Visual Studio Code development environment, and the MySQL database management system has been developed.

Scope of use – sale of goods via the Internet.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області	11
1.1 Системи опрацювання даних для продажу антикваріату	11
1.2 Аналіз систем опрацювання даних для продажу антикваріату	14
1.3 Висновки за розділом 1	23
2 Матеріали і методи	25
2.1 Вибір мови програмування	25
2.2 Вибір середовища розробки для створення системи опрацювання даних для продажу антикваріату	27
2.3 Вибір засобів зберігання та опрацювання даних	30
2.4 Висновки за розділом 2	33
3 Опис системи опрацювання даних	34
3.1 Структура системи опрацювання даних для продажу антикваріату	34
3.2 Функціонування системи опрацювання даних для продажу антикваріату	37
3.3 Розробка бази даних системи опрацювання даних для продажу антикваріату	40
3.4 Проєктування інтерфейсу системи опрацювання даних для продажу антикваріату	42
3.5 Висновки за розділом 3	44
4 Експлуатація, тестування та експериментальне дослідження системи опрацювання даних	46
4.1 Призначення й умови застосування системи опрацювання даних	46
4.2 Характеристики системи опрацювання даних для продажу антикваріату	48
4.3 Інструкція по експлуатації програми	49
4.3.1 Звернення до програми	49

4.3.2 Вхідні й вихідні дані	49
4.3.3 Повідомлення.....	49
4.4 Виконання системи опрацювання даних для продажу антикваріату	50
4.5 Тестування системи опрацювання даних для продажу антикваріату	53
4.6 Висновки за розділом 4	54
Висновки	55
Перелік джерел посилання	58
Додаток А Технічне завдання	60
Додаток Б Фрагмент тексту програми	64
Додаток В Слайди презентації	79

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

SQL – Structured Query Language;

БД – база даних;

ПА – продаж антикваріату;

СОД – система опрацювання даних.

ВСТУП

Зростання інтересу до ринку антикварних речей формує потребу в точному й надійному опрацюванні інформації про товар, його вартість, унікальні характеристики та історичне походження. У сфері продажу антикваріату обіг даних набуває особливої важливості, оскільки кожен об'єкт має індивідуальну цінність, що вимагає коректного зберігання описів, фотографій, підтверджувальних документів і результатів експертної перевірки. Застосування автоматизованих засобів дозволяє усунути залежність від ручної фіксації й мінімізувати ризик помилок під час роботи з рідкісними предметами [1], [2].

Потреба у швидкому доступі до інформації, можливості аналізувати ринкові тенденції та контролювати актуальність цін сприяє підвищенню попиту на сучасні програмні рішення. За наявності правильно структурованої програмної системи можна забезпечити прозорість торгових операцій, підвищити довіру покупців та полегшити процес визначення справжності товарів. Автоматизоване опрацювання даних та відповідні програмні системи створюють підґрунтя для розширення антикварних магазинів у віртуальному просторі, де інформація про цінності стає доступною ширшій аудиторії без втрати достовірності [3], [4].

Таким чином, система опрацювання даних у сфері продажу антикваріату розглядається як необхідний засіб підвищення якості обслуговування, оптимізації комерційних процесів і захисту унікальної культурної спадщини, що реалізується у вигляді товарів на ринку [1]-[7].

Проте, попри очевидну користь автоматизованих рішень, використання програмних систем для продажу антикваріату супроводжується низкою обмежень, що впливають на якість та достовірність інформаційного обігу. Через унікальний характер кожного предмета виникає складність у формалізованому описі характеристик, які часто потребують індивідуальної експертної оцінки, що не може бути повністю замінена алгоритмічними

механізмами. Ускладнення виникає й у питаннях зберігання та захисту документів, пов'язаних з походженням антикварних виробів. Недостатньо надійні системи можуть створювати ризик втрати важливих матеріалів або їх несанкціонованого зчитування. Крім того, програмні продукти потребують постійного технічного супроводу, оновлення та адаптації до змін ринку, що створює залежність від розробників і збільшує витрати на підтримку. Програмні рішення також можуть ускладнювати взаємодію з клієнтами, які надають перевагу традиційним методам купівлі, де значне значення мають особисті консультації та безпосередній огляд предметів. Таким чином, відомі програмні системи у сфері антикварної торгівлі не завжди забезпечують повну відповідність специфіці ринку та може спричиняти труднощі, що стосуються достовірності даних, безпеки інформації й збереження індивідуального підходу до оцінювання унікальних історичних цінностей [1]-[7].

Тому актуальною є розробка системи опрацювання даних для продажу антикваріату. У дипломній кваліфікаційній роботі бакалавра розв'язується актуальне завдання розробки системи опрацювання даних для продажу антикваріату для підвищення ефективності розповсюдження товарів.

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та систем опрацювання даних для продажу антикваріату;
- здійснити проєктування системи опрацювання даних для продажу антикваріату;
- створити систему опрацювання даних для продажу антикваріату;
- дослідити розроблену систему опрацювання даних для продажу антикваріату.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Системи опрацювання даних для продажу антикваріату

Система опрацювання даних для продажу антикваріату розглядається як комплексний інформаційний інструмент, що забезпечує реєстрацію, сортування, зберігання та використання відомостей про антикварні товари, їхню історичну цінність, походження, стан, цінові характеристики та потреби покупців. У подібних рішеннях управляється діяльність крамниць, аукціонів або комісійних пунктів, що працюють із рідкісними, старовинними або колекційними предметами. У таких системах відбувається постійне накопичення великої кількості описової інформації, стандартизація параметрів товарів, формування електронних карток кожної одиниці антикваріату та відстеження руху предметів, починаючи від моменту їх приймання й завершуючи продажем або поверненням власнику [1].

Застосування технологій опрацювання даних у сфері продажу антикваріату створює можливість формування точного цифрового обліку, підвищення прозорості операцій та покращення взаємодії між продавцями, покупцями, оцінювачами й постачальниками. Подібні системи здатні накопичувати велику базу знань про рідкісні предмети, що особливо важливо для виявлення підробок, визначення реальної вартості, забезпечення юридичної чистоти угод і мінімізації ризиків фінансового шахрайства. Завдяки автоматизації процесів із каталогу антикварних товарів зникає хаотичність, а ручний облік замінюється систематизованими діями, що допомагає уникати помилок у підрахунках, дублюванні даних або неправильному позначенні характеристик предметів. Наявність засобів аналітики сприяє аналізу попиту, прогнозуванню продажів та виявленню найбільш цінних груп товарів [2].

Однією з ключових переваг таких систем є можливість централізації всіх операцій. За допомогою загальної бази антикварні магазини, склади, аукціонні дома та комісійні відділи отримують доступ до узгоджених даних, що зменшує витрати часу на перевірку інформації та підвищує довіру

покупців. Іншою важливою перевагою вважається швидке формування оціночних звітів і експертних описів, які можуть бути представлені покупцеві в цифровому форматі або у вигляді експортованих документів. Невід'ємною складовою таких рішень є підтримка різноманітних інструментів пошуку та фільтрації, що дозволяє знаходити потрібні предмети за періодом створення, автором, матеріалом, категорією мистецтва, регіоном походження або станом збереження. Для великих колекцій цінним є також формування глибокої історичної хронології та прив'язки об'єктів до попередніх власників, майстерень, виробників або культурних епох [3].

Однак застосування таких систем супроводжується певними недоліками. Значні труднощі виникають у разі потреби створення надточних описів, що часто вимагає участі професійних експертів, оцінювачів та істориків мистецтва. Елементи автоматизації не здатні повністю замінити роботу людського фахівця, тому частина даних формується вручну і може містити суб'єктивність. Високі вимоги до безпеки даних та юридичних аспектів також створюють додаткові витрати, особливо під час міжнародних продажів або переміщення рідкісних предметів через кордони. Існує необхідність у належному захисті цифрових каталогів, адже втрата інформації про походження чи експертні висновки може завдати суттєвої шкоди бізнесу [4].

Особливості використання таких систем виражаються в специфіці товарних даних і потребі у високій точності опису. Кожен предмет антикваріату є унікальним, тому на відміну від звичайних торговельних рішень, де товари можуть мати тиражні характеристики, у системах антикварного продажу немає повторюваних одиниць. Це впливає на порядок ведення обліку, потребує ретельної фотографічної документації, визначення автентичності, оригінальності, іноді прив'язки до історичних архівів. Програми повинні підтримувати велику кількість параметрів для опису одного товару, і водночас забезпечувати швидкий доступ до всіх цих даних для оцінювачів, експертів і покупців [5].

У практиці використання таких технологій існує кілька видів систем. Одні функціонують як локальні платформи для окремих магазинів або аукціонів, забезпечуючи лише внутрішні потреби обліку. Інші реалізуються у форматі хмарних сервісів, що дозволяє обмінюватися інформацією між різними торговельними майданчиками або підключати сторонніх оцінювачів. Частина систем виступає як електронні ринки, де можуть реєструватися продавці та покупці, формуючи віртуальні вітрини антикваріату, брати участь в онлайн-аукціонах, здійснювати торги та укладати угоди дистанційно. Окремі рішення орієнтовані виключно на інвентаризацію та експертизу, не займаючись безпосереднім продажем, що робить їх частиною загальної технологічної інфраструктури ринку антикварних предметів [6].

Таким чином, системи опрацювання даних для продажу антикваріату є поєднанням цифрових технологій, експертної роботи та ринкової взаємодії. Вони забезпечують облік унікальних товарів, організацію продажів, встановлення достовірної цінності предметів історії та мистецтва, підтримуючи високий рівень безпеки та довіри під час комерційних операцій, проте залишаються залежними від людського аналізу, точності досліджень і складності правових та культурних аспектів, що супроводжують роботу з антикваріатом [6], [7].

Визначено, що система опрацювання даних для продажу антикваріату являє собою інформаційний інструмент, що забезпечує реєстрацію, сортування, зберігання та використання відомостей про антикварні товари, їхню історичну цінність, походження, стан, цінові характеристики та потреби покупців. У подібних програмних рішеннях підтримується діяльність крамниць, аукціонів або комісійних пунктів, що працюють із рідкісними, старовинними або колекційними предметами. У таких системах відбувається постійне накопичення великої кількості описової інформації, стандартизація параметрів товарів, формування електронних карток кожної одиниці антикваріату та відстеження руху предметів, починаючи від моменту їх приймання й завершуючи продажем або поверненням власнику.

1.2 Аналіз систем опрацювання даних для продажу антикваріату

Проаналізуємо системи опрацювання даних для продажу антикваріату [8]-[14].

Програмна система Cashier Live (рис. 1.1) є рішенням для організації торгівлі антикварними товарами, що функціонує через вебінтерфейс і дає змогу виконувати операції продажу, контролювати асортимент та формувати аналітичні дані про роботу магазину. Її принцип роботи ґрунтується на централізованому опрацюванні інформації, завдяки чому працівники можуть швидко актуалізувати дані про товари та проводити фінансові процедури з урахуванням специфіки антикварного ринку. Програмний продукт Cashier Live орієнтований на малий бізнес і створює можливість управляти унікальними позиціями, включаючи предмети, що реалізуються за домовленостями консигнації, коли товар надається власником для подальшого продажу [8], [12].

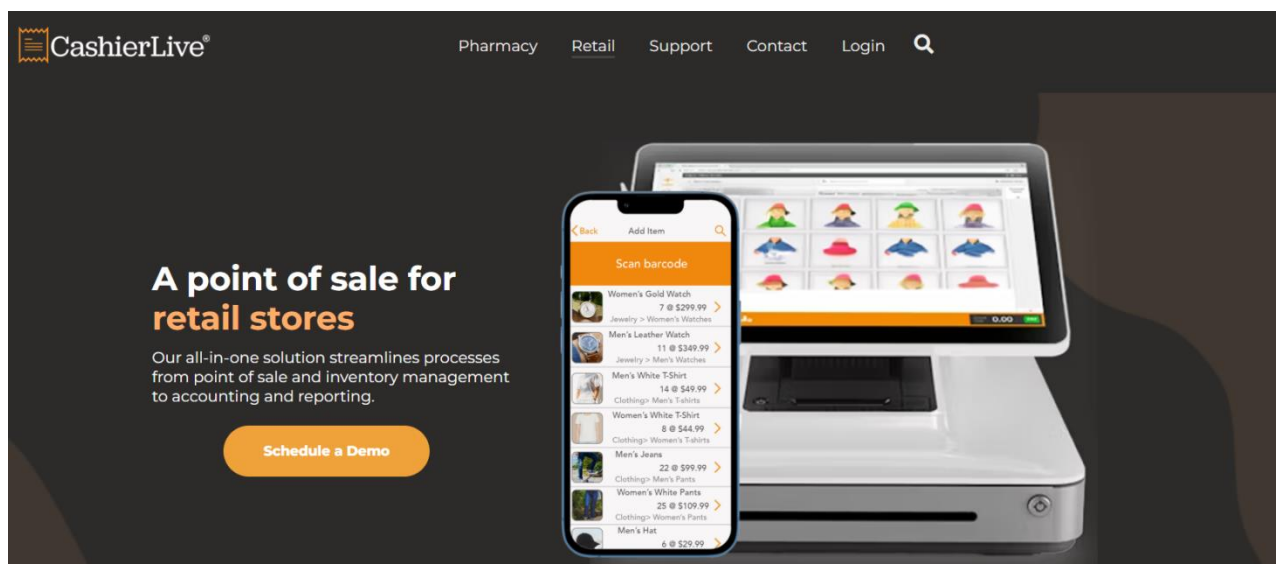


Рисунок 1.1 – Програмна система Cashier Live [12]

Програмна система Cashier Live підтримує ведення електронного обліку асортименту з прив'язкою до опису, фотографій, характеристик стану, експертних висновків та вартості. Передбачено засоби швидкого внесення та

змінення записів, включаючи сканування етикеток та створення документів, які супроводжують купівлю-продаж. Наявні механізми організації обліку клієнтів, перегляду історії взаємодій, застосування знижок, формування рахунків, а також електронного підпису й маркетингових інструментів. Підтримується робота з кількома торговими точками, тому управління запасами можливе одночасно для різних локацій [8], [12].

У системі передбачено функціональні можливості для контролю діяльності працівників, зокрема ведення інформації про виконані операції та робочий час. Інтегровані механізми опрацювання платежів забезпечують приймання банківських карток і прискорюють фінансові операції. Підприємство може отримувати дані про стан запасів і динаміку продажів у реальному часі завдяки зручній панелі керування [8], [12].

Попри широкі можливості, програмна система Cashier Live містить обмеження щодо взаємодії з додатковими сервісами та зовнішніми інструментами, що може звузити сферу застосування у випадках, коли потрібна розширена інтеграція з іншими системами. Водночас використання засобів миттєвого імпорту даних про товари дозволяє швидко формувати базу предметів і робить програму привабливою для магазинів, які працюють з унікальним антикваріатом та потребують точного обліку товарів, що перебувають на продажі за умовами консигнації [8], [12].

Основними характеристиками програмної системи Cashier Live є такі [8], [12]:

- веборієнтоване середовище: програмна система Cashier Live працює через Інтернет інтерфейс і дає змогу керувати продажами та товарами з будь-якого пристрою [8], [12];

- управління антикварним асортиментом: програмна система Cashier Live забезпечує внесення, редагування, пошук і відстеження товарів із деталізованими характеристиками та умовами консигнації [8], [12];

- облік клієнтів: програмна система Cashier Live підтримує збереження інформації про покупців, історії покупок та взаємодії з ними [8], [12];

- робота з декількома торговими точками: програмна система Cashier Live дає можливість централізовано контролювати запаси та продажі в різних магазинах [8], [12];
- управління персоналом: передбачено контроль дій працівників, облік робочого часу та відстеження операцій касирів [8], [12];
- генерація звітів: формуються аналітичні дані про продажі, запаси, динаміку доходів та ефективність роботи магазину [8], [12];
- інтегрована робота з платежами: у програмній системі Cashier здійснюється приймання банківських карток та електронна обробка транзакцій у касовому модулі [8], [12];
- імпорт товарів у реальному часі: миттєво завантажуються дані про предмети, що дозволяє швидко формувати інвентар і прискорювати роботу магазину [8], [12];
- підтримка електронних документів: у програмній системі Cashier створюються чеки, рахунки, рахунки-фактури та інші супровідні документи без паперових носіїв [8], [12];
- мобільні додатки: існують версії для смартфонів, що дозволяють працювати із системою поза касовим місцем [8], [12];
- програма лояльності: передбачено використання різних видів знижок та заохочень для постійних покупців [8], [12];
- електронний маркетинг: у програмній системі Cashier є можливість обробляти рекламні повідомлення та взаємодію з клієнтами в онлайн середовищі [8], [12].

Серед недоліків програмної системи Cashier Live можна відзначити обмежений набір інтеграцій із зовнішніми сервісами, що ускладнює використання додаткових інструментів для розширення функціоналу. Деякі модулі мають спрощений набір можливостей, тому користувач не завжди може налаштувати їх відповідно до специфічних потреб антикварного магазину. Також інтерфейс містить певні обмеження для детального редагування окремих параметрів товарів, особливо під час роботи з великим

обсягом даних. Крім того, розширені функції управління електронною комерцією або персоналізованою аналітикою можуть вимагати додаткових сервісів, які не підтримуються системою без сторонніх рішень [8], [12].

SimpleConsign (рис. 1.2) представляє собою веборієнтовану систему для автоматизації операцій з продажу та передачі товарів на комісію в антикварній сфері. Це програмне рішення орієнтується на спрощення взаємодії між магазинами та постачальниками, забезпечуючи дистанційний доступ до функцій додавання нових позицій, формування етикеток, контролю товарних залишків і керування ціновою політикою. Завдяки цьому постачальники можуть самостійно стежити за власним асортиментом і швидко вносити зміни без необхідності фізичної присутності [9], [11], [13].

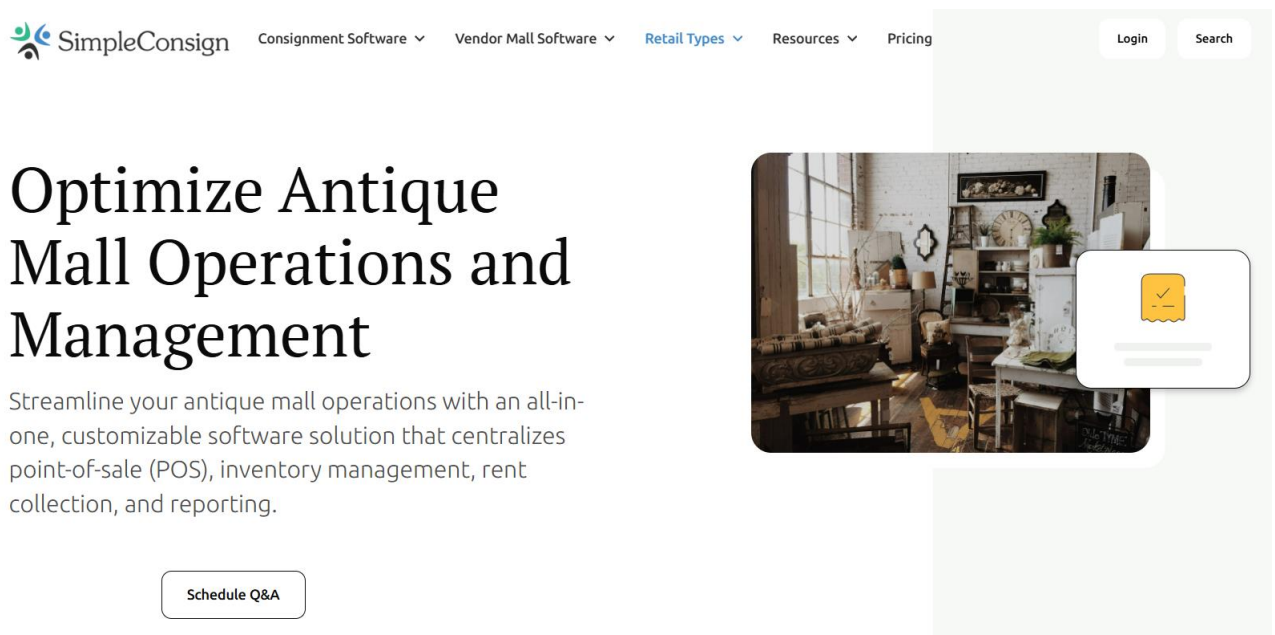


Рисунок 1.2 – Програмна система SimpleConsign [13]

Програмна система SimpleConsign дозволяє автоматизувати фінансові розрахунки з партнерами та відображає детальні аналітичні дані щодо продажів, що забезпечує власникам можливість швидко оцінювати ефективність роботи магазину. Програмний продукт підтримує інтеграцію з бухгалтерськими сервісами, що полегшує облік доходів і розрахунків. Окремою перевагою є можливість друкувати цінники на звичайному обладнанні, що

полегшує роботу невеликим торговим точкам. Інтерфейс відрізняється простотою, тому роботу із системою легко опанувати навіть користувачам без попереднього досвіду роботи з подібними рішеннями [9], [11], [13].

Програмна система SimpleConsign має такі особливості [9], [11], [13]:

- хмарна структура: програмна система SimpleConsign працює через вебінтерфейс і не потребує локальної установки на робоче місце, забезпечуючи доступ до даних у будь-який момент [9], [11], [13];

- дистанційна взаємодія постачальників: постачальники можуть самостійно додавати товари, друкувати штрих-коди та переглядати інформацію про продажі без прив'язки до магазину [9], [11], [13];

- автоматизовані розрахунки: програмна система SimpleConsign самостійно формує виплати постачальникам, що зменшує ризик помилок і прискорює фінансові операції [9], [11], [13];

- підтримка звітності: у програмній системі SimpleConsign передбачено формування аналітичних звітів у реальному часі, що дозволяє власникам магазину контролювати ефективність продажів [9], [11], [13];

- інтеграція з бухгалтерією: у програмній системі SimpleConsign передбачено взаємодію з обліковими інструментами для спрощення ведення бухгалтерського та фінансового обліку [9], [11], [13];

- друк етикеток без спеціального обладнання: програмна система SimpleConsign підтримує роботу зі звичайними принтерами, що зменшує витрати на оснащення магазину [9], [11], [13];

- зручність інтерфейсу: робота із програмною системою SimpleConsign не вимагає спеціальної підготовки, що спрощує адаптацію персоналу та постачальників [9], [11], [13].

Серед недоліків програмної системи SimpleConsign можна відзначити залежність від стабільного Інтернет з'єднання, що ускладнює роботу в умовах нестабільної мережі. Крім того, повноцінне використання функціоналу потребує постійної оплати за підписку, що може бути менш вигідним для невеликих антикварних магазинів з обмеженим бюджетом. Деякі користувачі

можуть також зіштовхнутися з обмеженнями у налаштуванні окремих функцій, що не дозволяє повністю адаптувати систему під специфічні бізнес-процеси конкретного підприємства [9], [11], [13].

GoAntiquing (рис. 1.3) позиціонується як програмна система, орієнтована саме на антикварні центри та крамниці, де продаж здійснюється через окремих продавців або власників кіосків. Система GoAntiquing працює на основі настільного термінала і створена для того, щоб забезпечити детальний контроль процесів, пов'язаних із реалізацією товарів, розрахунком орендної плати та комісійних відрахувань [9], [11], [14].

GoAntiquing! POS - [ADMIN]

File Edit Tools View Help

Cash Register

4 items, \$255.61 \$0.00 \$0.00 \$0.00 \$0.00 Corrections

New Sale TID: New sale Void

	Dealer*	ItemID	Qty	Description*	Price*	D%	SubTotal	Tax
1	14	100	1	LARGE LINEN TABLECLOTH	\$75.00	50	\$37.50	✓
2	14	200	12	VINTAGE LINEN NAPKINS	\$8.00	0	\$96.00	✓
3	3	99	1	PLATE SILVER SET	\$85.00		\$85.00	✓
4	3	99	2	VINTAGE DOILIES	\$8.00	0	\$16.00	✓

* Required information
4 items

SubTotal \$234.50
Tax \$21.11
Total \$255.61

☒ Cash
☐ Check
☐ Credit
☐ Debit
☐ Split
☐ Account
☐ New Layaway
☐ Reseller

Delete Item(s)
 Duplicate Item(s)

Last sale: \$39.24 (paid by check)

Complete Sale

My Antique Mall - C:\Data Files\GoAntiquing! POS\My Antique Mall.mdb CAPS NUM 04:02 PM

Рисунок 1.3 – Програмна система GoAntiquing [14]

За допомогою програмної система GoAntiquing продавці отримують можливість самостійно стежити за своїм товаром, формувати записи інвентарю та отримувати статистичні дані про продажі в електронному вигляді або через спеціальний вебпортал [9], [11], [14].

Функціональність програмної системи GoAntiquing охоплює роботу з різними видами платежів, підтримує структури комісій, ведення розрахунків за орендовані площі та обробку продажів у розстрочку. Крім того, у програмі передбачене автоматизоване створення резервних копій, які зберігаються у віддаленій інфраструктурі, що дозволяє власникам повністю захистити базу з даними про товари та операції. Інтерфейс програми GoAntiquing орієнтований не на стиль, а на практичне управління діяльністю підприємства, де важливо забезпечити доступ власників кіосків до звітності та контролю товарних залишків [9], [11], [14].

Програмна система GoAntiquing підходить для роботи з великою кількістю постачальників і моделей продажу, тому добре зарекомендувала себе серед підприємств, що спеціалізуються на реалізації колекційних речей і предметів антикваріату. Її особливістю є можливість детального відстеження взаємодії між власником торгового центру та окремими продавцями, що робить її зручною для середовищ, де велика увага приділяється роботі з постачальниками та управлінню їхньою діяльністю [9], [11], [14].

Програмна система GoAntiquing має такі особливості [9], [11], [14]:

- орієнтація на антикварний ринок: програмна система GoAntiquing створена спеціально для торгових центрів і магазинів, що реалізують колекційні та антикварні товари, із урахуванням специфіки роботи з дилерами та комітентами [9], [11], [14];

- детальний контроль постачальників: програмна система GoAntiquing забезпечує можливість управління комісійними відрахуваннями, відсотками оренди та правилами роботи кожного дилера без додаткових налаштувань до програми [9], [11], [14];

- робота з інвентарем продавців: дилери можуть самостійно формувати записи про товари, переглядати особисті показники продажів та відстежувати залишки через спеціальний вебпортал [9], [11], [14];

- автоматизоване резервне копіювання: інформація систематично зберігається в хмарній інфраструктурі програмної системи GoAntiquing, що дозволяє захистити велику базу даних і забезпечити відновлення у разі помилки чи втрати локальних даних [9], [11], [14];

- підтримка різних моделей обліку: програмна система GoAntiquing опрацьовує продажі з розстрочкою, розподілені платежі, індивідуальні правила оплати та формування фінансової звітності для окремих продавців товарів [9], [11], [14];

- електронна звітність: у програмній системі GoAntiquing результати продажів автоматично надсилаються продавцям електронною поштою або доступні онлайн у вигляді детальних звітів, що допомагає оперативно відстежувати прибутки [9], [11], [14];

- фокус на функціональності: програмна система GoAntiquing має простий інтерфейс, у якому головна увага приділяється управлінню торговими операціями, а не візуальним елементам, що спрощує використання у бізнесі з великою кількістю товарів і постачальників [9], [11], [14].

Серед недоліків програмного забезпечення GoAntiquing можна відзначити складність вивчення, оскільки для нових користувачів може знадобитися певний час на вивчення інтерфейсу та функціоналу Thinkific, особливо якщо вони раніше не користувалися подібними системами. Для деяких великих освітніх закладів або організацій, що обробляють великі обсяги даних, можуть виникати обмеження щодо зберігання та обробки інформації. Іншим недоліком є необхідність оплати за деякі функції [9], [14], що може викликати обмеження використання цієї програми невеликими освітніми установами або організаціями з обмеженим бюджетом [9], [11], [14].

Порівняльну характеристику систем опрацювання даних для продажу антикваріату наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння систем опрацювання даних для продажу антикваріату

Критерій порівняння	Cashier Live [12]	SimpleConsign [13]	GoAntiquing [14]
Хмарна архітектура	+	+	+—
Інструменти роботи з постачальниками	+—	+	+
Розширені можливості управління консигнаціями	+	+—	+—
Підтримка багатьох локацій (мультивітринність)	+	+—	+—
Інтеграція з бухгалтерськими системами	—	+	—
Невисока вартість використання	+—	+—	+

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато систем опрацювання даних для продажу антикваріату. Проте, попри очевидну користь автоматизованих рішень, використання програмних систем для продажу антикваріату супроводжується низкою обмежень, що впливають на якість та достовірність інформаційного обігу. Через унікальний характер кожного предмета виникає складність у формалізованому описі характеристик, які часто потребують індивідуальної експертної оцінки, що не може бути повністю замінена алгоритмічними механізмами. Ускладнення виникає й у питаннях зберігання та захисту документів, пов'язаних з походженням антикварних виробів. Недостатньо надійні системи можуть створювати ризик втрати важливих матеріалів або їх несанкціонованого зчитування. Крім того, програмні продукти потребують постійного технічного супроводу, оновлення та адаптації до змін ринку, що створює залежність від розробників і збільшує витрати на підтримку.

Програмні рішення також можуть ускладнювати взаємодію з клієнтами, які надають перевагу традиційним методам купівлі, де значне значення мають особисті консультації та безпосередній огляд предметів. Тому актуальною є розробка системи опрацювання даних для продажу антикваріату.

При розробці системи опрацювання даних для продажу антикваріату необхідно забезпечити такі функціональні вимоги:

- підтримка можливості подання інформації про антикварні товари;
- підтримка можливості вибору антикварних товарів за категорією;
- можливість ознайомлення з детальною інформацією про товар;
- підтримка функцій додавання обраних товарів до кошику;
- підтримка можливості роботи з кошиком товарів, зокрема, можливостей видання товарів;
- можливість оформлення замовлення на обрані товари.

1.3 Висновки за розділом 1

Визначено, що система опрацювання даних для продажу антикваріату являє собою інформаційний інструмент, що забезпечує реєстрацію, сортування, зберігання та використання відомостей про антикварні товари, їхню історичну цінність, походження, стан, цінові характеристики та потреби покупців. У подібних програмних рішеннях підтримується діяльність крамниць, аукціонів або комісійних пунктів, що працюють із рідкісними, старовинними або колекційними предметами.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато систем опрацювання даних для продажу антикваріату. Проте, попри очевидну користь автоматизованих рішень, використання програмних систем для продажу антикваріату супроводжується низкою обмежень, що впливають на якість та достовірність інформаційного обігу. Через унікальний характер кожного предмета виникає складність у формалізованому описі характеристик, які часто потребують індивідуальної

експертної оцінки, що не може бути повністю замінена алгоритмічними механізмами. Ускладнення виникає й у питаннях зберігання та захисту документів, пов'язаних з походженням антикварних виробів. Недостатньо надійні системи можуть створювати ризик втрати важливих матеріалів або їх несанкціонованого зчитування. Крім того, програмні продукти потребують постійного технічного супроводу, оновлення та адаптації до змін ринку, що створює залежність від розробників і збільшує витрати на підтримку. Програмні рішення також можуть ускладнювати взаємодію з клієнтами, які надають перевагу традиційним методам купівлі, де значне значення мають особисті консультації та безпосередній огляд предметів. Тому актуальною є розробка системи опрацювання даних для продажу антикваріату.

Сформульовано функціональні вимоги до системи опрацювання даних для продажу антикваріату.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці системи опрацювання даних для продажу антикваріату було використано мову програмування PHP [15], [16].

Мова програмування PHP вирізняється широкою екосистемою готових рішень, які значно прискорюють розроблення спеціалізованих систем. Велика кількість бібліотек і модулів дає змогу не створювати функціонал з нуля, а використовувати перевірені інструменти для керування користувачами, інтеграції платіжних сервісів, генерації електронних документів і безпечної передачі файлів. Такий підхід зменшує часові витрати на розроблення та дозволяє зосередитися на специфіці антикварного ринку [15], [16].

Мова програмування PHP також вирізняється високою сумісністю з більшістю вебсерверів, операційних систем і реляційних баз даних, що полегшує розгортання програмного забезпечення в різних інфраструктурних умовах. Програмну систему можна перенести на інший хостинг або сервер без суттєвих змін у коді, що підвищує гнучкість та довговічність рішення. Незалежність від конкретного апаратного чи програмного середовища дає змогу масштабувати продукт у разі зростання обсягів інформації або збільшення кількості користувачів [15], [16].

Окрему увагу заслуговує активна спільнота розробників, яка забезпечує постійну підтримку мови програмування PHP. Доступність навчальних матеріалів, оновлень і професійних консультацій спрощує розв'язання нетипових завдань, зокрема пов'язаних із перевіркою достовірності даних чи формуванням складних пошукових алгоритмів. Постійний розвиток мови та її фреймворків сприяє підвищенню продуктивності, безпеки та якості вебсервісів, що робить мову програмування PHP актуальним інструментом для створення сучасних програмних систем опрацювання даних, у тому числі для ринку антикваріату [15], [16].

Використання мови програмування PHP у розробленні системи опрацювання даних для продажу антикваріату вважається доцільним через поєднання технічної зручності та функціональних можливостей, що відповідають потребам такого спеціалізованого ринку. Оскільки торгівля антикварними предметами здебільшого здійснюється через вебплатформи, саме серверна мова, орієнтована на створення вебзастосунків, забезпечує стабільне керування даними, швидке формування сторінок, роботу з каталогами, зображеннями та документами, що підтверджують походження товарів [15], [16].

Мова програмування PHP дозволяє ефективно обробляти бази даних, підтримує роботу з великими обсягами описів, а також дає можливість реалізувати модулі автентифікації, оцінювання та керування історією об'єктів. Наявність широкого набору фреймворків полегшує впровадження захисту від несанкціонованого доступу, що є важливим у разі зберігання експертних висновків, сертифікаційних документів та унікальних фотографічних матеріалів. Гнучкість мови програмування PHP допомагає розробникам адаптувати систему до специфічних функцій, зокрема інструментів пошуку за рідкісними критеріями, перевірки унікальності або відстеження ринкової вартості [15], [16].

Значною перевагою стає й економічна доступність PHP, оскільки для його використання не потрібні дорогі ресурси чи складні налаштування серверних платформ. Це дає можливість створювати ефективні рішення навіть на обмеженому бюджеті, що особливо важливо для невеликих антикварних магазинів або приватних продавців. Завдяки поєднанню веборієнтованості, високої сумісності з базами даних та невисокої вартості впровадження, PHP забезпечує оптимальні умови для розроблення спеціалізованої системи опрацювання даних у сфері продажу антикваріату [15], [16].

Обґрунтування вибору мови програмування для розробки системи опрацювання даних для продажу антикваріату наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки системи опрацювання даних для продажу антикваріату

Критерій порівняння мов програмування	Мова програмування		
	PHP	JavaScript	Ruby
Швидкість розроблення	+	+—	+—
Зручність для вебплатформ	+	+	+—
Простота інтеграції з базами даних	+	+—	—
Масштабованість	+	+—	+
Наявність готових бібліотек і модулів	+	+	+—

Отже, для реалізації системи опрацювання даних для продажу антикваріату обрано мову програмування PHP, яка дозволяє ефективно обробляти бази даних, підтримує роботу з великими обсягами описів, а також дає можливість реалізувати модулі автентифікації, оцінювання та керування історією об'єктів. Крім того, мова PHP забезпечує стабільне керування даними, швидке формування сторінок, роботу з каталогами, зображеннями та документами, що підтверджують походження товарів.

2.2 Вибір середовища розробки для створення системи опрацювання даних для продажу антикваріату

В якості середовища розробки для створення системи опрацювання даних для продажу антикваріату було обрано середовище Visual Studio Code [17], [18].

Середовище розробки Visual Studio Code вирізняється зручними інструментами налагодження, які дають змогу відстежувати помилки в реальному часі, аналізувати обробку даних та контролювати взаємодію з

базами даних чи API сервісів, що можуть використовуватися для перевірки автентичності товарів. Інтелектуальні підказки, підтримка автодоповнення та глибока інтеграція із системами тестування допомагають зменшити кількість технічних помилок, що особливо важливо в системах, де обробляються точні та рідкісні дані [17], [18].

Середовище розробки Visual Studio Code може адаптуватися до потреб різних команд розробників. Завдяки підтримці хмарної синхронізації налаштувань і розширень забезпечується можливість продовжувати розроблення на різних пристроях без втрати середовища роботи. Для проєктів, пов'язаних із продажем антикваріату, така мобільність дозволяє швидко реагувати на зміни, оновлювати модулі, додавати інформацію про нові експонати або коригувати алгоритми оцінювання незалежно від місця перебування розробника [17], [18].

Завдяки відкритому коду Visual Studio Code підтримується активна спільнота, яка постійно розробляє нові плагіни, оптимізує роботу редактора й розширює його функціональні можливості. Постійний розвиток інструменту сприяє тому, що середовище залишається актуальним навіть для нестандартних застосунків, наприклад, систем зі складною логікою оцінювання антикварних предметів, інтегрованими експертними модулями або підсистемами машинного аналізу даних [17], [18].

Використання Visual Studio Code для створення системи опрацювання даних у сфері продажу антикваріату вважається доцільним через поєднання гнучкості, доступності та широких можливостей налаштування. Це середовище не вимагає значних ресурсів комп'ютера, швидко працює навіть на середньому обладнанні та підтримує різні інструменти, необхідні для веброзроблення та опрацювання інформації про унікальні предмети [17], [18].

Завдяки великому набору розширень у Visual Studio Code можна нешвидко налаштувати роботу з мовами програмування, базами даних, різними форматами документів і зображень, які часто використовуються для фіксації характеристик антикварних виробів. Інтеграція з системами контролю

версій спрощує супровід проєкту, дозволяє відстежувати зміни та забезпечує безпечне зберігання коду, що особливо важливо під час створення систем, які оперують цінною інформацією [17], [18].

Зручність роботи з мережею плагінів у Visual Studio Code дає можливість автоматизувати перевірку типів даних, виявляти помилки та покращувати безпеку, уникаючи загроз несанкціонованого доступу до конфіденційних документів і зображень антикварних предметів. Середовище підтримує роботу з фреймворками, що допомагають створювати функції пошуку, каталогізації, експертної оцінки та онлайн торгівлі [17], [18].

Обґрунтування вибору середовища розробки для створення системи опрацювання даних для продажу антикваріату наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення системи опрацювання даних для продажу антикваріату

Критерій порівняння середовищ розробки	Середовища розробки		
	Visual Studio Code	PhpStorm	Aptana Studio
Можливість гнучкого розширення	+	+—	+—
Підтримка плагінів для баз даних	+	+	+—
Інструменти налагодження і тестування	+	+	—
Підтримка роботи з вебфреймворками	+	+	+—
Низькі вимоги до обладнання	+	—	+—

Таким чином, Visual Studio Code створює умови для ефективної, безпечної та технологічно гнучкої розробки спеціалізованої інформаційної системи, забезпечуючи комфортну підтримку усіх етапів реалізації — від написання коду до впровадження та подальшого адміністрування [17], [18].

Отже, для створення системи опрацювання даних для продажу антикваріату обрано середовище розробки Visual Studio Code, що поєднує швидкодію, безплатність, максимальну гнучкість та наявність великої кількості розширень для роботи з PHP, зображеннями, базами даних, контролем версій і захистом інформації.

2.3 Вибір засобів зберігання та опрацювання даних

Для зберігання та опрацювання даних було обрано систему керування базами даних MySQL [19], [20].

Система керування базами даних MySQL підтримує складні типи запитів, що дає змогу виконувати аналітичні операції над інформацією. Це корисно під час формування статистики продажів, розрахунку популярності певних стилів, оцінювання попиту на конкретні категорії товарів або визначення динаміки змін вартості антикварних виробів. Такі можливості сприяють не лише обліку, а й прогнозуванню тенденцій ринку [19], [20].

Важливою особливістю системи керування базами даних MySQL є можливість інтеграції з хмарними сервісами, що забезпечує дистанційне зберігання інформації, резервне копіювання та надійний захист від втрати даних навіть у випадку виходу з ладу локального обладнання. Резервне копіювання дозволяє швидко відновити базу та уникнути ризиків, які можуть виникнути при зберіганні унікальних описів антикварних предметів або експертних документів [19], [20].

Отже система керування базами даних MySQL не лише підтримує роботу із структурованою інформацією, але й створює умови для безпечного доступу, поглибленої аналітики, адаптації до вимог ринку та надійного

захисту даних, що робить її особливо корисною для систем опрацювання даних у сфері антикварної торгівлі [19], [20].

Використання системи керування базами даних MySQL для створення системи опрацювання даних у сфері продажу антикваріату вважається доцільним через поєднання надійності, швидкодії та зручності роботи зі складними інформаційними структурами. У базах даних такого типу зберігаються описи предметів, фотографії, відомості про походження, експертні висновки, результати оцінювання, документи та історія продажів, що потребують чіткої організації та гарантованої цілісності. Система керування базами даних MySQL забезпечує захист взаємопов'язаних даних, уникає можливості їхньої втрати під час збоїв та підтримує транзакційність, що важливо для операцій купівлі та продажу [19], [20].

Система керування базами даних MySQL вирізняється високою швидкістю опрацювання запитів, що дозволяє миттєво формувати каталоги, виконувати пошук за рідкісними характеристиками та фільтрувати товари за історичними періодами, матеріалами, стилями або підтверджувальними документами. Надійність MySQL у роботі з великими обсягами текстової інформації та файлових посилань сприяє точному відображенню унікальних властивостей антикварних виробів [19], [20].

Значну перевагу становить безкоштовність MySQL та її сумісність із найпоширенішими мовами програмування і вебсерверами. Це дає змогу реалізувати проєкт без істотних фінансових витрат і спростити інтеграцію з вебплатформою, де виконується продаж. Система легко масштабується, тому база даних може розширюватися разом із розвитком магазину: зростанням кількості товарів, відгуків користувачів, додаванням експертних модулів та інструментів оцінювання [19], [20].

Таким чином, MySQL створює оптимальне середовище для зберігання й швидкого опрацювання структурованих і ціннісно важливих даних, забезпечує безпечне виконання торгових операцій та надає економічно вигідне підґрунтя для розвитку системи продажу антикваріату [19], [20].

Обґрунтування вибору засобів зберігання та опрацювання даних наведено у таблиці 2.3.

Таблиця 2.3 – Обґрунтування вибору засобів зберігання та опрацювання даних

Критерій порівняння систем управління базами даних	Системи управління базами даних		
	MySQL	PostgreSQL	Oracle Database
Швидкість обробки запитів	+	+—	+—
Простота налаштування та розгортання	+	+—	—
Інтеграція з вебтехнологіями	+	+	+—
Невисока вартість використання	+	+	+—
Простота резервного копіювання	+	+	+—
Активність спільноти	+	+—	—

Таким чином, для зберігання та опрацювання даних обрано систему управління базами даних MySQL, яка поєднує високу швидкість, стабільність, легкість розгортання та безкоштовність, що робить її оптимальною для веборієнтованих систем опрацювання даних, у тому числі для торгівлі антикваріатом. Крім того, вона підтримує інтеграцію з вебплатформами, масштабованість та надійність збереження інформації, що забезпечує ефективну роботу з каталогами, документами, фотографіями та історією продажів унікальних предметів.

2.4 Висновки за розділом 2

Для реалізації системи опрацювання даних для продажу антикваріату обрано мову програмування PHP, яка дозволяє ефективно обробляти бази даних, підтримує роботу з великими обсягами описів, а також дає можливість реалізувати модулі автентифікації, оцінювання та керування історією об'єктів. Крім того, мова PHP забезпечує стабільне керування даними, швидке формування сторінок, роботу з каталогами, зображеннями та документами, що підтверджують походження товарів.

Для створення системи опрацювання даних для продажу антикваріату обрано середовище розробки Visual Studio Code, що поєднує швидкодію, безплатність, максимальну гнучкість та наявність великої кількості розширень для роботи з PHP, зображеннями, базами даних, контролем версій і захистом інформації.

Для зберігання та опрацювання даних обрано систему управління базами даних MySQL, яка поєднує високу швидкість, стабільність, легкість розгортання та безкоштовність, що робить її оптимальною для веборієнтованих систем опрацювання даних, у тому числі для торгівлі антикваріатом. Крім того, вона підтримує інтеграцію з вебплатформами, масштабованість та надійність збереження інформації, що забезпечує ефективну роботу з каталогами, документами, фотографіями та історією продажів унікальних предметів.

3 ОПИС СИСТЕМИ ОПРАЦЮВАННЯ ДАНИХ

3.1 Структура системи опрацювання даних для продажу антикваріату

Структуру системи опрацювання даних для продажу антикваріату наведено на рис. 3.1.



Рисунок 3.1 – Структура системи опрацювання даних для продажу антикваріату

Структура системи опрацювання даних для продажу антикваріату передбачає взаємодію кількох ключових модулів, кожен з яких відповідає за окремий сегмент функціональності та забезпечує повноцінну роботу вебплатформи. Така структура забезпечує безпечне внесення даних про товари, можливість оновлення й редагування каталогу, організацію замовлень, опрацювання відгуків покупців та формування адміністративного інтерфейсу. Водночас вся логіка роботи скоординована таким чином, щоб користувач отримував коректну інформацію, а адміністратор міг керувати Інтернет магазином швидко й без зайвих дій.

Модуль роботи з каталогом та категоризації антикваріату відповідає за створення, структурування та формування груп товарів, що відрізняються за стилем, епохою, походженням або природою матеріалів. Основу його роботи складає можливість визначати окремі категорії, до яких належать різні антикварні предмети, а також редагувати їх, видаляти або створювати нові при розширенні асортименту. Такий модуль забезпечує логічну класифікацію товарів, що полегшує навігацію як для користувача, так і для адміністратора. Його функціональна складова охоплює перевірку коректності введених даних, роботу з назвами таблиць бази даних, встановлення властивостей категорій та подальше збереження цієї інформації.

Модуль управління товарами та їх описами забезпечує наповнення сайту реальними об'єктами антикваріату. Його призначення полягає в додаванні нових товарів, редагуванні характеристик, заповненні описів, зміні фотографій, а також видаленні товарів після продажу або в разі відсутності необхідності демонстрації. У межах цього модуля у системі опрацювання даних для продажу антикваріату здійснюється контроль змісту картки товару, який включає історичну характеристику, стан збереження, походження, фотогалерею та стартову вартість. Модуль управління товарами та їх описами також відповідає за перевірку правильності внесених даних, структурування таблиці з товарами та збереження інформації до бази даних.

Модуль замовлень та оформлення покупок призначений для організації процесу придбання антикварних предметів. Його функції охоплюють створення нових замовлень, запис інформації про покупця, перевірку даних, фіксацію стану замовлень та передачу їх на опрацювання адміністратором. У межах модуля замовлень та оформлення покупок виконуються операції з транзакційністю даних, що дозволяє уникати помилок під час збереження інформації про купівлю. Також реалізується перевірка введених даних, що особливо важливо в умовах роботи з цінними предметами, оскільки некоректне оформлення замовлення може призвести до фінансових втрат або спотворення інформації про товар.

Модуль відгуків та комунікації служить засобом для збору суб'єктивних оцінок від покупців, які взаємодіяли з товаром або здійснювали придбання. Такий модуль підсилює довіру до ресурсу, допомагає іншим користувачам орієнтуватися під час вибору та формує репутацію продавця на основі попередніх угод. У структурі цього модуля закладено механізм перевірки контенту, запобігання публікації некоректних або неправдивих відгуків, а також систему збереження коментарів у базі даних. Реалізуються перевірка атрибутів, застосування правил валідації та внесення відгуків за допомогою функцій збереження.

Модуль адміністративного керування магазином забезпечує роботу адміністратора над усіма складовими системи. Цей модуль передбачає доступ до окремої панелі керування, де можна переглядати нові й архівні замовлення, здійснювати редагування товарів, категорій, аналізувати відгуки та видаляти некоректні дані. У межах цього модуля реалізовано механізми переходу між сторінками керування, завантаження інтерфейсу редагування, формування списків товарів та повний контроль за торговою платформою. Адміністративний модуль фактично координує роботу всіх інших модулів, здійснюючи вплив на функціональність каталогу, замовлень та контенту, який бачить користувач.

Модуль користувацького інтерфейсу та навігації спрямований на взаємодію з покупцем, який переглядає каталог, обирає предмети та звертається до розділів із контактами або інформацією про сайт. Він забезпечує завантаження головних сторінок, перегляд товарів певної категорії, ознайомлення з описом окремого товару, формування заявки на замовлення та отримання контактної інформації. У межах цього модуля реалізовано логіку показу сторінок, візуальне структурування даних, вибудовування зрозумілого маршруту для користувача, що дозволяє спростити придбання антикваріату.

Таким чином, система опрацювання даних для продажу антикваріату складається з комплексної сукупності модулів, що виконують різні функції, але утворюють єдиний цілісний механізм. Їхня взаємодія забезпечує можливість повноцінного управління контентом, швидкої обробки замовлень, підтримання репутації через відгуки та зручного керування платформою як для адміністратора, так і для пересічного користувача.

3.2 Функціонування системи опрацювання даних для продажу антикваріату

Функціонування системи опрацювання даних для продажу антикваріату подамо за допомогою схеми, зображеної на рис. 3.2.

Функціонування системи опрацювання даних для продажу антикваріату спрямоване на безперервну взаємодію користувача, адміністратора та сховища даних, де зберігається інформація про колекційні товари. Під час її роботи виконуються послідовні програмні процедури, які забезпечують перегляд каталогу, керування асортиментом, обробку замовлень та доступ адміністратора до внутрішніх функцій системи. Уся логіка побудована на перевірці введених даних, формуванні запитів до бази та збереженні змін, що гарантує достовірність і впорядкованість інформації.

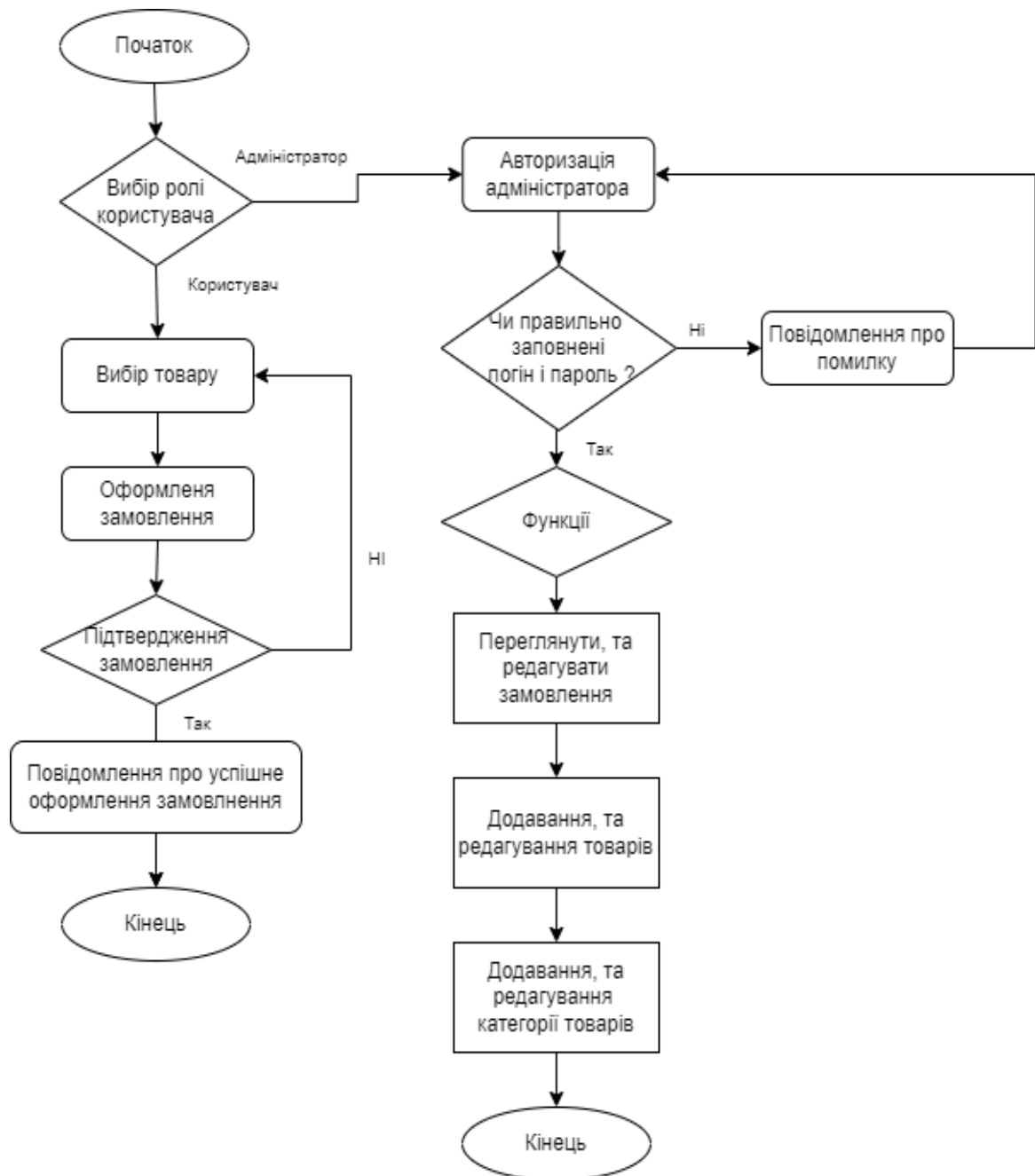


Рисунок 3.2 – Схема функціонування системи опрацювання даних для продажу антикваріату

Для роботи адміністратора системи опрацювання даних для продажу антикваріату передбачено спеціальний процес входу до системи. Спершу здійснюється заповнення полів авторизаційної форми, після чого дані перевіряються на відповідність тим, що зберігаються у базі. Якщо інформація збігається з записами, сервер надає доступ до адміністративної панелі. У випадку помилки користувач не допускається до закритої частини сервісу.

Таким чином, робота з товарним каталогом та замовленнями можлива лише після підтвердження особи адміністратора.

Процес придбання товарів з боку покупця відбувається через спеціальну форму. Після введення контактної інформації та вибору предмета система перевіряє заповнення обов'язкових полів, аналізує правильність даних і лише після цього вносить замовлення до бази. Якщо під час перевірки виявляються неточності, користувач отримує повідомлення про помилку і має змогу виправити дані. Таким чином, оформлення покупки відбувається лише після підтвердження правильності запиту.

Робота з товарами у системі передбачає як створення нового запису, так і зміну вже наявного. При додаванні товару адміністратор вводить необхідну інформацію через форму, після чого програма перевіряє її на відповідність правилам. Якщо вимоги дотримано, антикварний предмет стає частиною каталогу. Для редагування існуючої позиції потрібно знайти її у списку, оновити значення і підтвердити зміни, які система заносить до бази лише після повторної перевірки коректності.

Схожим способом функціонує механізм роботи з категоріями товарів. Спочатку адміністратор додає назву та властивості нової категорії, а система контролює правильність даних. У разі успішної перевірки створюється новий запис. Якщо є потреба в корекції, визначається потрібна категорія, оновлюються її характеристики та фіксуються зміни в базі. Такі процедури забезпечують впорядкованість структури каталогу та точність прив'язки товарів до відповідних розділів.

Загалом система опрацювання даних для продажу антикваріату працює за єдиним принципом: кожна дія користувачів чи адміністратора фіксується лише після перевірки введених значень і успішного зв'язку з базою. Такий підхід гарантує узгодженість даних, усуває ризики появи неповної або недостовірної інформації та створює надійну основу для роботи платформи продажу антикварних предметів.

3.3 Розробка бази даних системи опрацювання даних для продажу антикваріату

База даних, що використовується у системі продажу антикварних предметів, побудована таким чином, щоб забезпечувати повне і структуроване зберігання інформації про товари, замовлення, категорії та відгуки покупців. Вона складається з кількох взаємопов'язаних таблиць, кожна з яких відповідає за окремий тип даних, а зв'язки між ними дозволяють формувати повний опис кожного антикварного об'єкта разом з коментарями та покупцями, які цікавилися цим товаром.

Схему бази даних системи опрацювання даних для продажу антикваріату наведено на рис. 3.3.

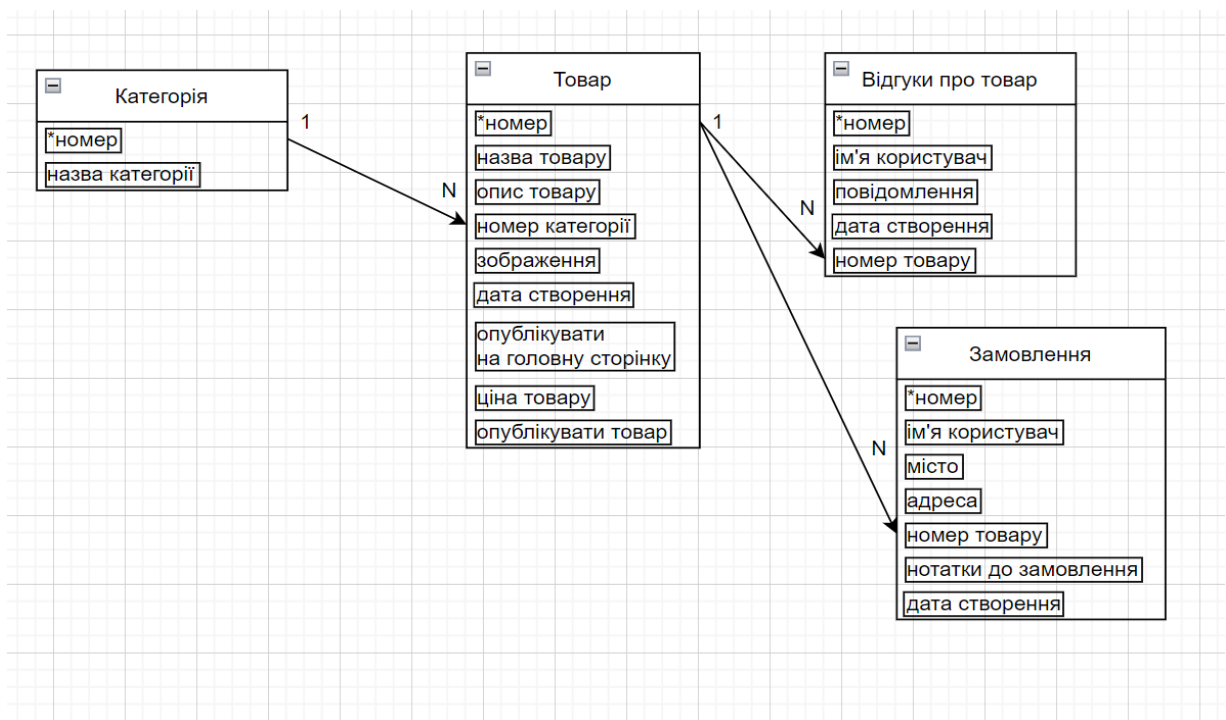


Рисунок 3.3 – Схема бази даних системи опрацювання даних для продажу антикваріату

Основна інформація про антикварні предмети міститься у таблиці, призначеній для фіксації характеристик кожного товару. У ній зберігаються назва, опис, категорія, графічне зображення, ціна, дата додавання до каталогу,

а також службові параметри, що дозволяють визначати актуальність товару та можливість його відображення на головній сторінці. Ідентифікація кожного елемента відбувається за допомогою числового унікального коду. Уся структурована інформація про товар пов'язується з іншими таблицями через поле, яке використовується для встановлення зв'язків між товаром і відповідними категоріями чи замовленнями.

Окрему роль у структурі відіграє таблиця категорій, яка дозволяє розділяти антикварні предмети за групами. До неї включені ідентифікатор та назва категорії. Така таблиця є допоміжною та використовується в інших сутностях для того, щоб кожен товар мав чітко визначену тематичну належність. Завдяки цій таблиці можливе формування каталогу, фільтрація товарів і сортування їх за групами.

Щоб забезпечити можливість додавання відгуків до предметів, у базі існує таблиця, в якій фіксуються коментарі користувачів. Тут містяться ім'я автора відгуку, його текст, дата написання та посилання на товар, щодо якого залишено коментар. Саме через це посилання система може виводити всі відгуки, що стосуються конкретного предмета, роблячи каталог більш інформативним і корисним для користувача.

Для збереження замовлень передбачена окрема таблиця, що містить дані покупця та інформацію про обрану річ. У ній знаходяться ім'я замовника, спосіб доставки, його місцезнаходження, адреса отримання товару, додаткові позначки щодо замовлення та ідентифікатор обраного антикварного виробу. Збереження цих даних дозволяє системі автоматично обробляти замовлення, пов'язувати їх із конкретними товарами та формувати подальші дії адміністратора.

Завдяки поділу інформації на декілька взаємопов'язаних таблиць забезпечується нормалізована структура бази даних. Така побудова дає можливість зменшити дублювання даних, підвищити швидкість виконання запитів і створити основу для подальшого розширення функціоналу системи продажу антикваріату без порушення логіки її роботи.

3.4 Проєктування інтерфейсу системи опрацювання даних для продажу антикваріату

Проєктування інтерфейсу взаємодії користувача з системою опрацювання даних для продажу антикваріату виконано з урахуванням вимог технічного завдання. При розробленні системи опрацювання даних для продажу антикваріату враховані функціональні вимоги до програми:

- підтримка можливості подання інформації про антикварні товари;
- підтримка можливості вибору антикварних товарів за категорією;
- можливість ознайомлення з детальною інформацією про товар;
- підтримка функцій додавання обраних товарів до кошику;
- підтримка можливості роботи з кошиком товарів, зокрема, можливостей видання товарів;
- можливість оформлення замовлення на обрані товари.

Інтерфейс головної форми системи опрацювання даних для продажу антикваріату наведено на рис. 3.4.

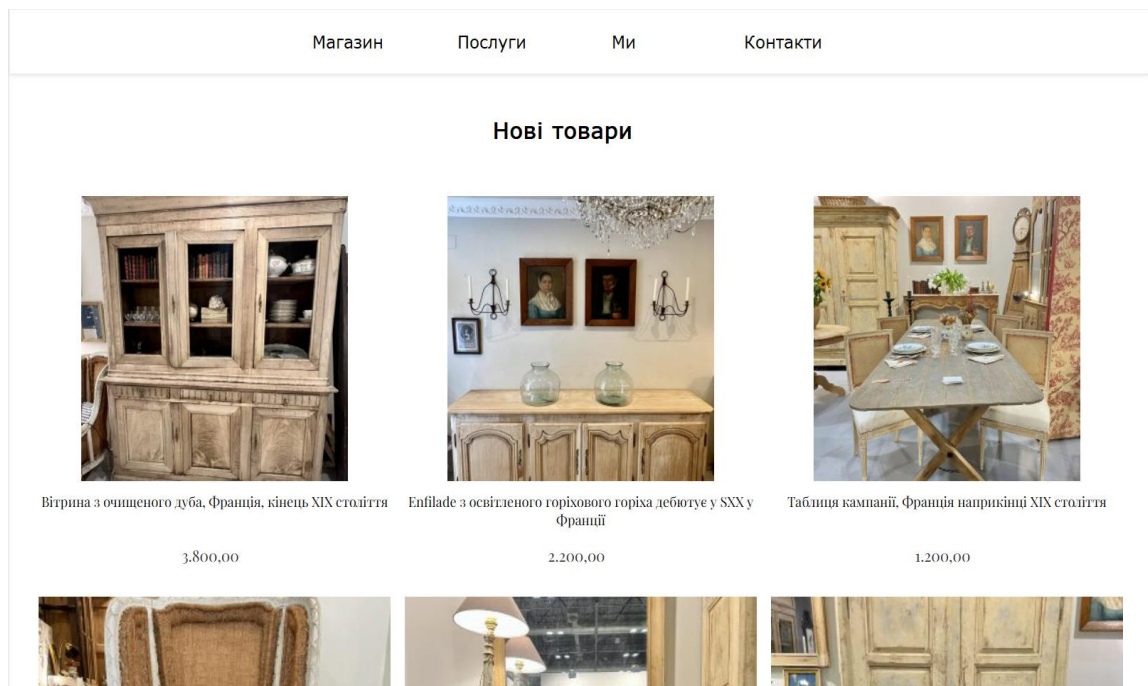


Рисунок 3.4 – Інтерфейс головної форми

Інтерфейс прототипу головного меню наведено на рис. 3.5. Сторінку з прикладом опису товару наведено на рис. 3.6. Сторінку кошику товарів наведено на рис. 3.7.

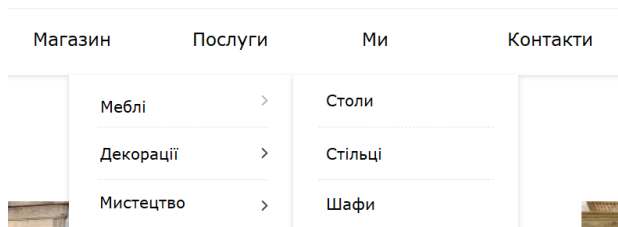


Рисунок 3.5 – Інтерфейс прототипу головного меню

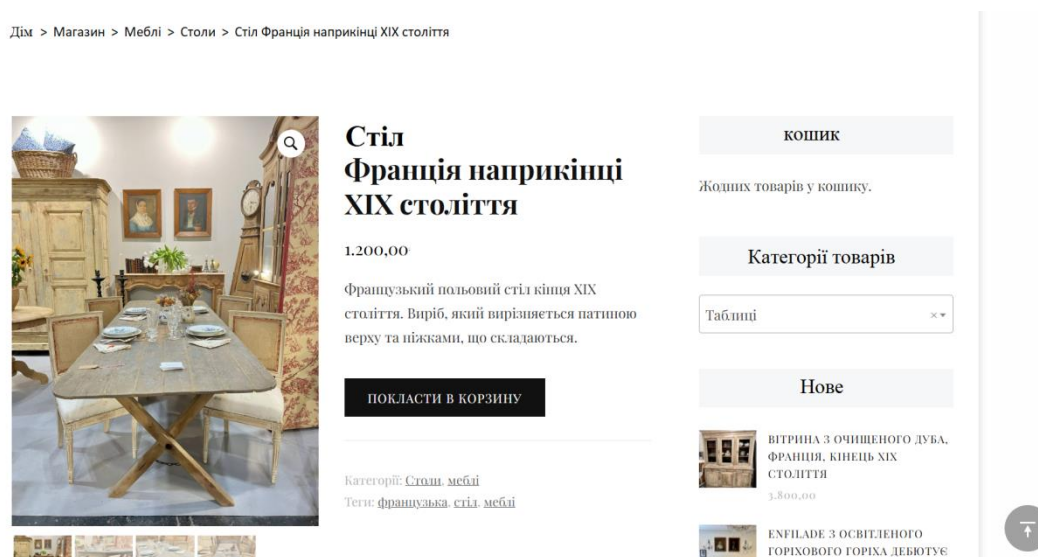


Рисунок 3.6 – Сторінка з прикладом опису товару

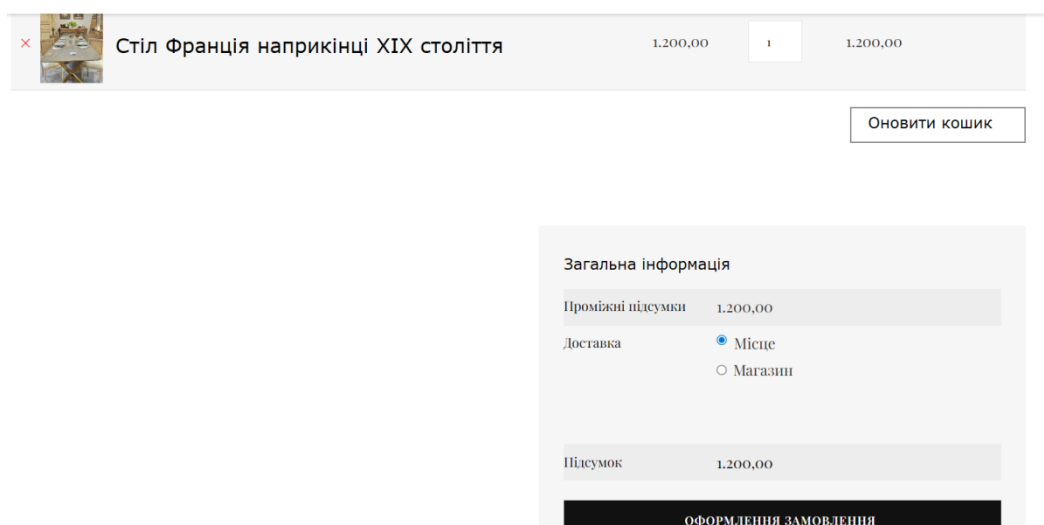


Рисунок 3.7 – Сторінка кошику товарів

Прототип форми оформлення замовлення наведено на рис. 3.8.

Рисунок 3.8 – Форма оформлення замовлення

3.5 Висновки за розділом 3

Розроблено систему опрацювання даних для продажу антикваріату.

Запропоновано структуру системи опрацювання даних для продажу антикваріату. Визначено, що структура системи опрацювання даних для продажу антикваріату передбачає взаємодію кількох ключових модулів, кожен з яких відповідає за окремий сегмент функціональності та забезпечує повноцінну роботу вебплатформи. Така структура забезпечує безпечне внесення даних про товари, можливість оновлення й редагування каталогу, організацію замовлень, опрацювання відгуків покупців та формування адміністративного інтерфейсу.

Описано функціонування системи опрацювання даних для продажу антикваріату, що спрямоване на безперервну взаємодію користувача, адміністратора та сховища даних, де зберігається інформація про колекційні товари. Під час роботи виконуються послідовні програмні процедури, які забезпечують перегляд каталогу, керування асортиментом, обробку замовлень

та доступ адміністратора до внутрішніх функцій системи. Уся логіка побудована на перевірці введених даних, формуванні запитів до бази та збереженні змін, що гарантує достовірність і впорядкованість інформації.

Розроблено базу даних системи опрацювання даних для продажу антикваріату, яка побудована таким чином, щоб забезпечувати повне і структуроване зберігання інформації про товари, замовлення, категорії та відгуки покупців. Вона складається з кількох взаємопов'язаних таблиць, кожна з яких відповідає за окремий тип даних, а зв'язки між ними дозволяють формувати повний опис кожного антикварного об'єкта разом з коментарями та покупцями, які цікавилися цим товаром.

Виконано проєктування інтерфейсу взаємодії користувача з системою опрацювання даних для продажу антикваріату.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ СИСТЕМИ ОПРАЦЮВАННЯ ДАНИХ

4.1 Призначення й умови застосування системи опрацювання даних

Система опрацювання даних призначена для підтримки процесу продажу антикваріату. Система опрацювання даних написана на мові PHP, яка дозволяє ефективно обробляти бази даних, підтримує роботу з великими обсягами описів, а також дає можливість реалізувати модулі автентифікації, оцінювання та керування історією об'єктів. В якості середовища розробки для створення системи опрацювання даних для продажу антикваріату обрано Visual Studio Code, що поєднує швидкодію, безплатність, максимальну гнучкість та наявність великої кількості розширень для роботи з PHP, зображеннями, базами даних, контролем версій і захистом інформації. Для зберігання та опрацювання даних обрано систему управління базами даних MySQL, яка поєднує високу швидкість, стабільність, легкість розгортання та безкоштовність, що робить її оптимальною для веборієнтованих систем опрацювання даних, у тому числі для торгівлі антикваріатом.

Система опрацювання даних для продажу антикваріату забезпечує виконання таких функцій:

- підтримка можливості подання інформації про антикварні товари;
- підтримка можливості вибору антикварних товарів за категорією;
- можливість ознайомлення з детальною інформацією про товар;
- підтримка функцій додавання обраних товарів до кошику;
- підтримка можливості роботи з кошиком товарів, зокрема, можливостей видання товарів;
- можливість оформлення замовлення на обрані товари.

Для роботи системи опрацювання даних для продажу антикваріату потрібні такі технічні та програмні ресурси: комп'ютерна інфраструктура повинна забезпечувати стабільну обробку вебзапитів, швидке збереження

даних і безперебійну роботу бази. На апаратному рівні бажано використовувати процесор із тактовою частотою не нижче 2,0 ГГц, оскільки цього достатньо для виконання серверних обчислень та обробки декількох запитів одночасно. Обсяг оперативної пам'яті повинен становити не менше 4 ГБ, що дозволяє системі працювати з масивами товарів, коментарів і замовлень без затримок. З боку програмного забезпечення система повинна працювати в середовищі, де встановлений сучасний вебсервер на зразок Apache або Nginx, що забезпечуватиме стабільну роботу вебзастосунку та обробку запитів користувачів. Обов'язковою є підтримка мови програмування PHP версії не нижче 7.4, оскільки саме на ній реалізована логіка роботи інтерфейсів та взаємодія з базою даних. Для збереження структурованих даних у системі використовується MySQL, тому на сервері має бути встановлена відповідна СКБД з можливістю керування таблицями, зв'язками та транзакціями. Додатково бажана підтримка бібліотек і модулів PHP, необхідних для роботи з графікою, валідацією форм та обміном даними зі сторінками адміністративної панелі.

Для розвитку і підтримки програмного коду необхідно наявність середовища розробки, яке підтримує формат проекту, автодоповнення синтаксису і роботу з PHP-фреймворками. Найзручнішим варіантом є Visual Studio Code з установленими розширеннями для роботи з PHP, MySQL, HTML та CSS. Також система вимагатиме веббраузера актуальної версії для тестування інтерфейсу, перегляду сторінок та контролю правильності відображення елементів каталогу і функцій замовлення. Завдяки цим вимогам забезпечується комплексне, стабільне та безпечне функціонування сервісу продажу антикваріату.

Функціональні характеристики розробленої системи опрацювання даних для продажу антикваріату наведено у технічному завданні (додаток А). Фрагмент тексту програмного коду системи опрацювання даних для продажу антикваріату наведено у додатку Б.

4.2 Характеристики системи опрацювання даних для продажу антикваріату

Система опрацювання даних для продажу антикваріату характеризується здатністю забезпечувати повний цикл взаємодії між товаром, покупцем та адміністратором ресурсу. Вона підтримує обробку інформації про антикварні предмети, управління замовленнями, ведення категорій та формування відгуків, залишених користувачами. Завдяки структурованому поділу даних обробка здійснюється послідовно та без втрати інформації, причому кожен об'єкт каталогу має власні атрибути та зв'язки із супутніми сутностями.

Система здатна оптимізувати роботу адміністратора за рахунок автоматизованого додавання, редагування та видалення товарів і категорій, дозволяючи контролювати наявність об'єктів у каталозі та коригувати їх статус. У процесі взаємодії з користувачами вона підтримує оформлення замовлення та передачу інформації до бази даних, завдяки чому покупець отримує можливість залишити запит на придбання обраного предмета. Підтримується робота з коментарями, що дає змогу збільшити інформативність каталогу, підвищити довіру до товарів і забезпечити реальний зворотний зв'язок.

Система демонструє стійкість до одночасного опрацювання кількох запитів, здатність працювати у вебсередовищі та адаптацію до зростання обсягів збережених даних. Вона підтримує збереження медіафайлів, забезпечуючи можливість відображення зображень антикварних об'єктів разом з описом. Функціонування реалізоване таким чином, щоб інтерфейси для адміністратора та користувача залишалися інтуїтивними й не потребували спеціальних навичок для виконання основних дій.

Система опрацювання даних для продажу антикваріату також вирізняється гнучкістю подальшого розширення: структура бази та архітектура модулів дозволяють у майбутньому додавати додаткові функції,

не порушуючи цілісності вже наявного функціоналу. Усе це забезпечує стабільність, доступність та ефективність застосування ресурсу у сфері електронної торгівлі антикварними товарами.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для звертання до системи опрацювання даних для продажу антикваріату необхідно перейти на відповідну вебсторінку.

4.3.2 Вхідні й вихідні дані

Вхідними даними до системи опрацювання даних для продажу антикваріату є інформація, що надходить від користувача під час виконання певних дій. У систему передаються відомості про товари, включаючи назву, опис, зображення, вартість та категорію, а також дані про потенційного покупця, які потрібні для оформлення замовлення.

Вихідними даними системи опрацювання даних для продажу антикваріату є відображені результати обробки введеної інформації у вигляді сформованих сторінок каталогу, переглядів товарів, створених замовлень і опублікованих відгуків.

4.3.3 Повідомлення

Користувач системи опрацювання даних для продажу антикваріату може отримати такі повідомлення: оповіщення про успішне оформлення замовлення, повідомлення про необхідність заповнення всіх обов'язкових полів або про некоректність введених даних, застереження щодо відсутності товару. Такі повідомлення допомагають уникнути помилок, надають

користувачеві чітко розуміння виконаних дій та сприяють коректній взаємодії з системою.

4.4 Виконання системи опрацювання даних для продажу антикваріату

Після запуску системи опрацювання даних для продажу антикваріату користувачу відображається головна сторінка (рис. 4.1), де знаходиться головне меню, а також область з останніми надходженнями антикварних товарів.

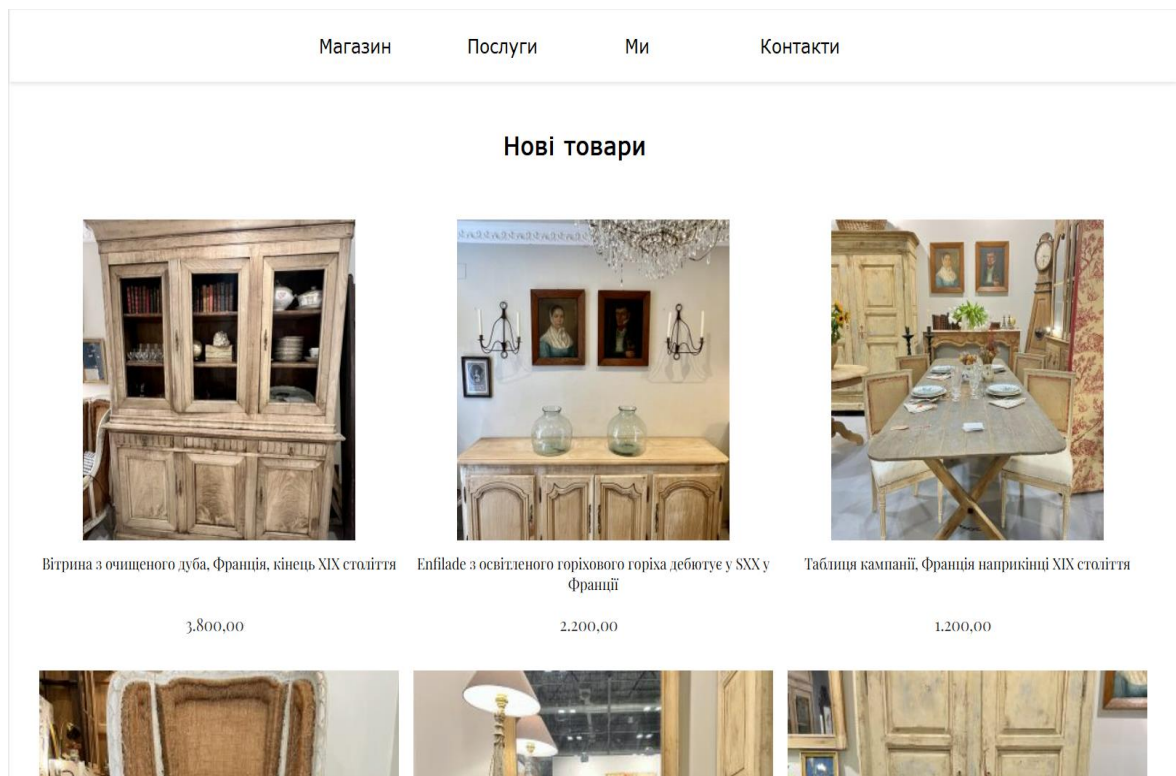


Рисунок 4.1 – Інтерфейс головної сторінки системи опрацювання даних для продажу антикваріату

Зверху головної сторінки користувач може бачити головне меню (рис. 4.2).

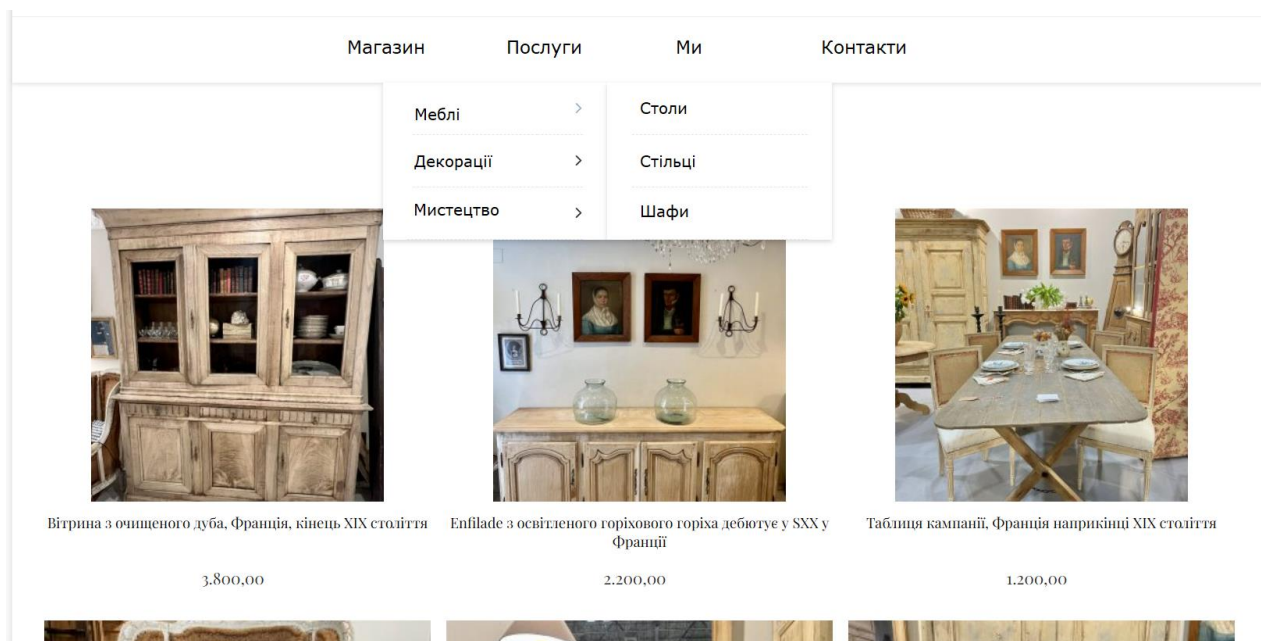


Рисунок 4.2 – Головне меню

При виборі предмету антикваріату відбувається перехід до відповідної сторінки (рис. 4.3), де міститься інформація про обраний предмет, його ціну, категорію та ін.

Дім > Магазин > Меблі > Столи > Стіл Франція наприкінці XIX століття



Стіл Франція наприкінці XIX століття

1.200,00

Французький польовий стіл кінця XIX століття. Виріб, який вирізняється патиною верху та ніжками, що складаються.

ПОКЛАСТИ В КОРЗИНУ

Категорії: [Столи](#), [меблі](#)

Теги: [французька](#), [стіл](#), [меблі](#)

КОШИК

Жодних товарів у кошику.

Категорії товарів

Таблиці

Нове



ВІТРИНА З ОЧИЩЕНОГО ДУБА,
ФРАНЦІЯ, КІНЕЦЬ XIX
СТОЛІТТЯ
3.800,00



ENFILADE З ОСВІТЛЕНОГО
ГОРІХОВОГО ГОРІХА ДЕБЮТУЄ

Рисунок 4.3 – Інформація про обраний предмет антикваріату

У випадку бажання придбання обраного предмету користувач може покласти його у кошик покупок (рис. 4.4).

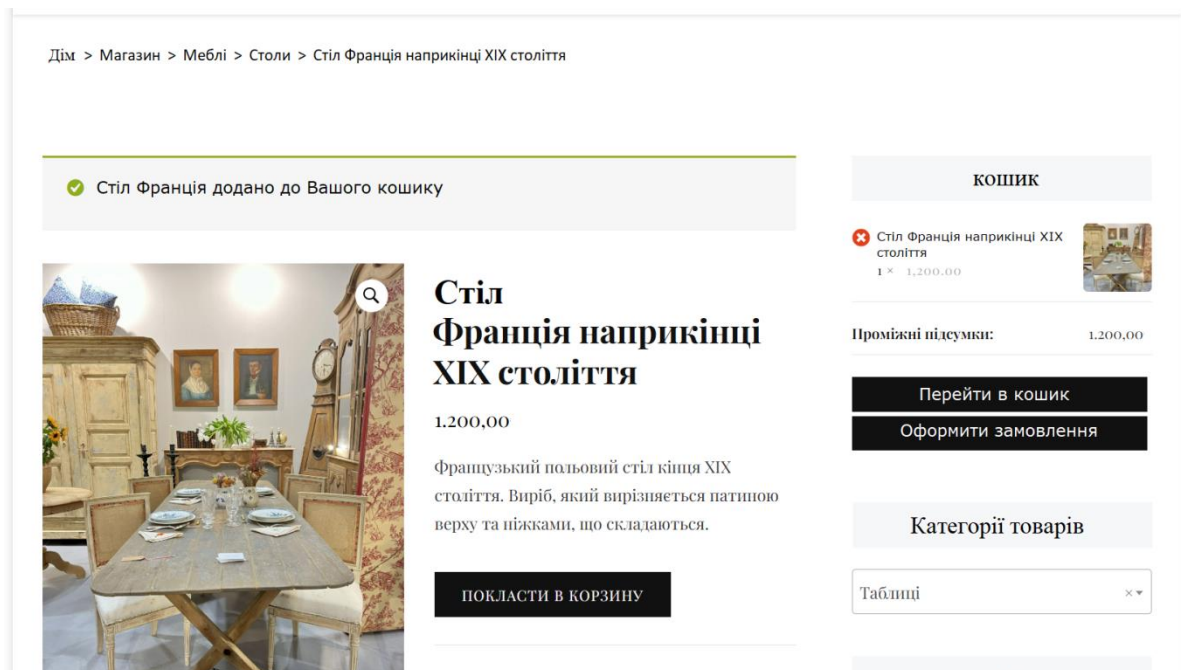


Рисунок 4.4 – Додавання обраного предмету до кошику покупок

При переході до кошику відображається список обраних товарів з ціною (рис. 4.5).

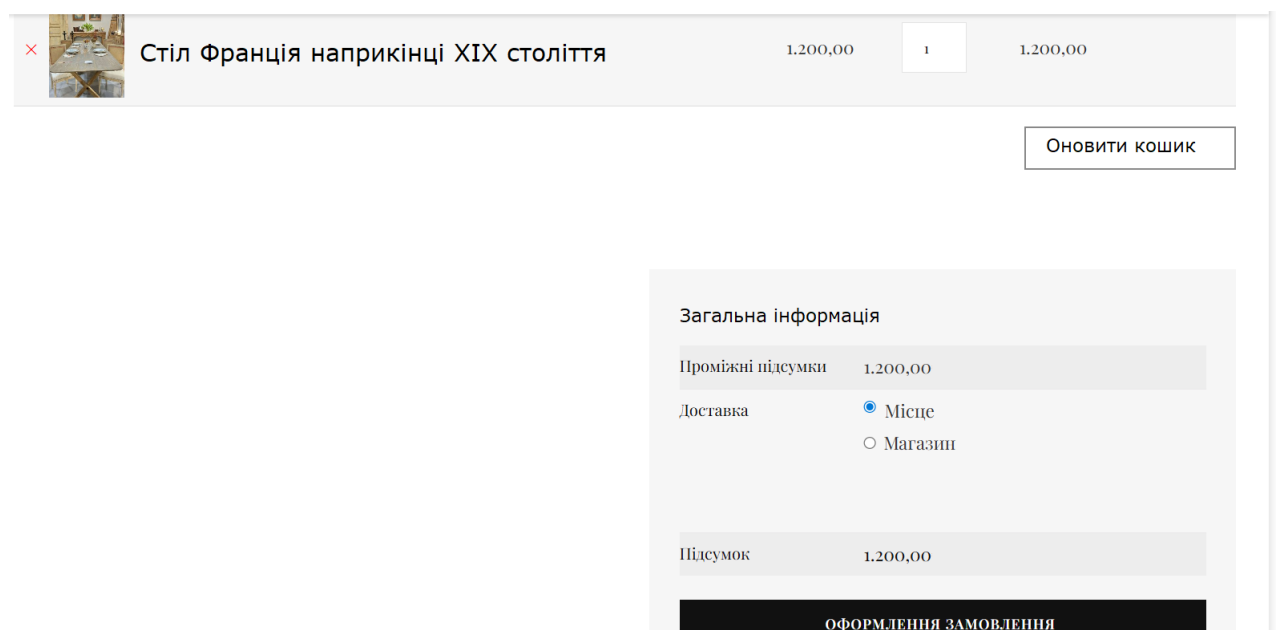


Рисунок 4.5 – Кошик товарів

Для оформлення замовлення необхідно натиснути відповідну кнопку та у формі, що з'явилася (рис. 4.6), ввести дані для отримання (прізвище та ім'я, адреса) та обрати спосіб оплати.

Рисунок 4.6 – Форма оформлення замовлення

4.5 Тестування системи опрацювання даних для продажу антикваріату

Проведено тестування системи опрацювання даних для продажу антикваріату.

Підтверджено, що системи опрацювання даних для продажу антикваріату працює коректно і стабільно, виконуючи всі необхідні функціональні вимоги та успішно справляючись із основною задачею підтримки процесу продажу товарів.

Розроблена система забезпечує організацію та опрацювання даних, пов'язаних із підтримкою процесів продажу антикварних товарів.

4.6 Висновки за розділом 4

Описано систему опрацювання даних для продажу антикваріату.

Проведено тестування створеної системи опрацювання даних для продажу антикваріату. Результати тестування показали, що система забезпечує організацію та опрацювання даних, пов'язаних із підтримкою процесів продажу антикварних товарів.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процеси обчислень, пов'язані з організацією та опрацюванням даних для продажу антикваріату.

Визначено, що система опрацювання даних для продажу антикваріату являє собою інформаційний інструмент, що забезпечує реєстрацію, сортування, зберігання та використання відомостей про антикварні товари, їхню історичну цінність, походження, стан, цінові характеристики та потреби покупців. У подібних програмних рішеннях підтримується діяльність крамниць, аукціонів або комісійних пунктів, що працюють із рідкісними, старовинними або колекційними предметами.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато систем опрацювання даних для продажу антикваріату. Проте, попри очевидну користь автоматизованих рішень, використання програмних систем для продажу антикваріату супроводжується низкою обмежень, що впливають на якість та достовірність інформаційного обігу. Через унікальний характер кожного предмета виникає складність у формалізованому описі характеристик, які часто потребують індивідуальної експертної оцінки, що не може бути повністю замінена алгоритмічними механізмами. Ускладнення виникає й у питаннях зберігання та захисту документів, пов'язаних з походженням антикварних виробів. Недостатньо надійні системи можуть створювати ризик втрати важливих матеріалів або їх несанкціонованого зчитування. Крім того, програмні продукти потребують постійного технічного супроводу, оновлення та адаптації до змін ринку, що створює залежність від розробників і збільшує витрати на підтримку. Програмні рішення також можуть ускладнювати взаємодію з клієнтами, які надають перевагу традиційним методам купівлі, де значне значення мають особисті консультації та безпосередній огляд предметів. Тому актуальною є розробка системи опрацювання даних для продажу антикваріату.

Сформульовано функціональні вимоги до системи опрацювання даних для продажу антикваріату.

Для реалізації системи опрацювання даних для продажу антикваріату обрано мову програмування PHP, яка дозволяє ефективно обробляти бази даних, підтримує роботу з великими обсягами описів, а також дає можливість реалізувати модулі автентифікації, оцінювання та керування історією об'єктів. Крім того, мова PHP забезпечує стабільне керування даними, швидке формування сторінок, роботу з каталогами, зображеннями та документами, що підтверджують походження товарів.

Для створення системи опрацювання даних для продажу антикваріату обрано середовище розробки Visual Studio Code, що поєднує швидкодію, безплатність, максимальну гнучкість та наявність великої кількості розширень для роботи з PHP, зображеннями, базами даних, контролем версій і захистом інформації.

Для зберігання та опрацювання даних обрано систему управління базами даних MySQL, яка поєднує високу швидкість, стабільність, легкість розгортання та безкоштовність, що робить її оптимальною для веборієнтованих систем опрацювання даних, у тому числі для торгівлі антикваріатом. Крім того, вона підтримує інтеграцію з вебплатформами, масштабованість та надійність збереження інформації, що забезпечує ефективну роботу з каталогами, документами, фотографіями та історією продажів унікальних предметів.

Розроблено систему опрацювання даних для продажу антикваріату.

Запропоновано структуру системи опрацювання даних для продажу антикваріату. Визначено, що структура системи опрацювання даних для продажу антикваріату передбачає взаємодію кількох ключових модулів, кожен з яких відповідає за окремий сегмент функціональності та забезпечує повноцінну роботу вебплатформи. Така структура забезпечує безпечне внесення даних про товари, можливість оновлення й редагування каталогу,

організацію замовлень, опрацювання відгуків покупців та формування адміністративного інтерфейсу.

Описано функціонування системи опрацювання даних для продажу антикваріату, що спрямоване на безперервну взаємодію користувача, адміністратора та сховища даних, де зберігається інформація про колекційні товари. Під час роботи виконуються послідовні програмні процедури, які забезпечують перегляд каталогу, керування асортиментом, обробку замовлень та доступ адміністратора до внутрішніх функцій системи. Уся логіка побудована на перевірці введених даних, формуванні запитів до бази та збереженні змін, що гарантує достовірність і впорядкованість інформації.

Розроблено базу даних системи опрацювання даних для продажу антикваріату, яка побудована таким чином, щоб забезпечувати повне і структуроване зберігання інформації про товари, замовлення, категорії та відгуки покупців. Вона складається з кількох взаємопов'язаних таблиць, кожна з яких відповідає за окремий тип даних, а зв'язки між ними дозволяють формувати повний опис кожного антикварного об'єкта разом з коментарями та покупцями, які цікавилися цим товаром.

Виконано проєктування інтерфейсу взаємодії користувача з системою опрацювання даних для продажу антикваріату.

Проведено тестування створеної системи опрацювання даних для продажу антикваріату. Результати тестування показали, що система забезпечує організацію та опрацювання даних, пов'язаних із підтримкою процесів продажу антикварних товарів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Getting Started with an Antique Mall Point of Sale Software: A Beginner's Guide. URL: <https://www.connectpos.com/antique-mall-point-of-sale-software/> (date of access: 19.04.2026).
2. Upgrade To Modern Antique Business Management. URL: <https://www.antiquetrack.com/> (date of access: 19.04.2026).
3. Point-of-Sale Software for Antique Centers and Emporiums. URL: <https://www.antique-ly.com/> (date of access: 19.04.2026).
4. Use a Business Management Software for your antiques shop services. URL: <https://www.blackbell.com/en/antiques-shop-business-management-software> (date of access: 19.04.2026).
5. Buy, Sell Antiques Online Website. URL: <https://www.thehoarde.com/> (date of access: 19.04.2026).
6. Antique Selling Tips That Won't Make You Look Like a Total Rookie. URL: <https://decofond.com/antique-selling-tips/> (date of access: 19.04.2026).
7. Ways to Sell Your Antiques for the Most Money. URL: <https://declutteringmom.com/15-ways-to-sell-your-antiques-for-the-most-money/> (date of access: 19.04.2026).
8. Antique Store Inventory Software: 3 Best Systems. URL: <https://wisessmallbusiness.com/antique-store-inventory-software/> (date of access: 19.04.2026).
9. Best POS Systems for Antique Malls to Create a Delightful Shopping Experience. URL: <https://www.connectpos.com/best-pos-systems-for-antique-malls/> (date of access: 19.04.2026).
10. Software for Antique, Art and Collectible Dealers. URL: <https://www.aain.com/aainsoftwareproducts.html> (date of access: 19.04.2026).
11. Best Small Business Consignment Software. URL: <https://sourceforge.net/software/consignment/for-small-business/> (date of access: 19.04.2026).

12. Cashier Live. URL: <https://cashierlive.com/retail> (date of access: 19.04.2026).

13. SimpleConsign. URL: [https:// www.simpleconsign.com/antique-malls](https://www.simpleconsign.com/antique-malls) (date of access: 19.04.2026).

14. GoAntiquing. URL: <https://www.goantiquing.net/> (date of access: 19.04.2026).

15. PHP-Handbuch. URL: <https://www.php.net/manual/de/> (date of access: 19.04.2026).

16. PHP For Windows. URL: <https://windows.php.net/download/> (date of access: 19.04.2026).

17. PhpStorm. Empowering PHP Developers. URL: <https://www.jetbrains.com/phpstorm/promo/> (date of access: 19.04.2026).

18. Unlocking Productivity with PhpStorm. URL: <https://medium.com/@AlexanderObregon/unlocking-productivity-with-phpstorm-a-beginners-guide-ea54c1b12cea> (date of access: 19.04.2026).

19. Introduction to MySQL. URL: https://www.w3schools.com/mysql/mysql_intro.asp (date of access: 19.04.2026).

20. MySQL HeatWave. URL: <https://dev.mysql.com/doc/> (date of access: 19.04.2026).

ДОДАТОК А
Технічне завдання

Вступ

Програмна система опрацювання даних може використовуватися для підтримки процесу продажу антикварних товарів.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Створення програмної системи опрацювання даних для продажу антикваріату», затверджене наказом Національного університету «Запорізька політехніка» № 139 від 7 квітня 2026 р.

A.2 Призначення розробки

Система опрацювання даних призначена для забезпечення продажу антикваріату.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Система опрацювання даних для продажу антикваріату забезпечує виконання таких функцій:

- підтримка можливості подання інформації про антикварні товари;
- підтримка можливості вибору антикварних товарів за категорією;
- можливість ознайомлення з детальною інформацією про товар;
- підтримка функцій додавання обраних товарів до кошику;
- підтримка можливості роботи з кошиком товарів, зокрема, можливостей видання товарів;
- можливість оформлення замовлення на обрані товари.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс системи опрацювання даних для продажу антикваріату повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

А.3.3 Вимоги до надійності

Система опрацювання даних для продажу антикваріату повинна забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації системи опрацювання даних для продажу антикваріату необхідна наявність персонального комп'ютера.

А.3.5 Вимоги до складу та параметрів технічний засобів

Для роботи системи опрацювання даних для продажу антикваріату потрібні такі технічні та програмні ресурси: комп'ютерна інфраструктура повинна забезпечувати стабільну обробку вебзапитів, швидке збереження даних і безперебійну роботу бази. На апаратному рівні бажано використовувати процесор із тактовою частотою не нижче 2,0 ГГц, оскільки цього достатньо для виконання серверних обчислень та обробки декількох запитів одночасно. Обсяг оперативної пам'яті повинен становити не менше 4 ГБ, що дозволяє системі працювати з масивами товарів, коментарів і замовлень без затримок. З боку програмного забезпечення система повинна працювати в середовищі, де встановлений сучасний вебсервер на зразок

Apache або Nginx, що забезпечуватиме стабільну роботу вебзастосунку та обробку запитів користувачів. Обов'язковою є підтримка мови програмування PHP версії не нижче 7.4, оскільки саме на ній реалізована логіка роботи інтерфейсів та взаємодія з базою даних. Для збереження структурованих даних у системі використовується MySQL, тому на сервері має бути встановлена відповідна СКБД з можливістю керування таблицями, зв'язками та транзакціями. Додатково бажана підтримка бібліотек і модулів PHP, необхідних для роботи з графікою, валідацією форм та обміном даними зі сторінками адміністративної панелі.

A.3.6 Вимоги до маркування і пакування

Система опрацювання даних для продажу антикваріату може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва – «Система опрацювання даних для продажу антикваріату».

A.4 Вхідні дані до роботи

Вхідними даними до системи опрацювання даних для продажу антикваріату є інформація, що надходить від користувача під час виконання певних дій. У систему передаються відомості про товари, включаючи назву, опис, зображення, вартість та категорію, а також дані про потенційного покупця, які потрібні для оформлення замовлення.

Вихідними даними системи опрацювання даних для продажу антикваріату є відображені результати обробки введеної інформації у вигляді сформованих сторінок каталогу, переглядів товарів, створених замовлень і опублікованих відгуків.

ДОДАТОК Б
Фрагмент тексту програми

```

class Config {
    public static function get() {
        return [
            'db_host' => 'localhost',
            'db_name' => 'antq_db',
            'db_user' => 'antq_usr',
            'db_pass' => 'antq_pwd',
            'upload_dir' => __DIR__ . '/uploads'
        ];
    }
}

class DB {
    private $pdo;
    private $cfg;
    public function __construct($cfg) {
        $this->cfg = $cfg;
    }
    public function connect() {
        if ($this->pdo !== null) return $this->pdo;
        $dsn = "mysql:host={$this->cfg['db_host']};dbname={$this->cfg['db_name']};charset=utf8mb4";
        $opt = [
            \PDO::ATTR_ERRMODE => \PDO::ERRMODE_EXCEPTION,
            \PDO::ATTR_DEFAULT_FETCH_MODE => \PDO::FETCH_ASSOC,
            \PDO::ATTR_EMULATE_PREPARES => false
        ];
        $this->pdo = new \PDO($dsn, $this->cfg['db_user'], $this->cfg['db_pass'], $opt);
        return $this->pdo;
    }
    public function fetch($sql, $params = []) {
        $st = $this->connect()->prepare($sql);
        $st->execute($params);
        return $st->fetch();
    }
    public function fetchAll($sql, $params = []) {
        $st = $this->connect()->prepare($sql);
        $st->execute($params);
        return $st->fetchAll();
    }
    public function exec($sql, $params = []) {
        $st = $this->connect()->prepare($sql);
        $st->execute($params);
        return $st;
    }
    public function insert($table, $data) {

```

```

        $cols = array_keys($data);
        $fields = implode('`,`', $cols);
        $place = implode(',', array_map(function($c){ return ':';
. $c; }, $cols));
        $sql = "INSERT INTO `{ $table}` (`{ $fields}`) VALUES
({ $place})";
        $this->exec($sql, $data);
        return $this->connect()->lastInsertId();
    }
    public function update($table, $data, $whereCol, $whereVal) {
        $sets = [];
        foreach ($data as $k => $v) $sets[] = "`$k` = :$k";
        $setStr = implode(',', $sets);
        $data['_whr'] = $whereVal;
        $sql = "UPDATE `{ $table}` SET { $setStr} WHERE
`{ $whereCol}` = :_whr";
        $this->exec($sql, $data);
        return true;
    }
    public function delete($table, $col, $val) {
        $sql = "DELETE FROM `{ $table}` WHERE `{ $col}` = :v";
        $this->exec($sql, ['v' => $val]);
        return true;
    }
}

class Utils {
    public static function json($data, $code = 200) {
        http_response_code($code);
        header('Content-Type: application/json; charset=utf-8');
        echo json_encode($data, JSON_UNESCAPED_UNICODE);
        exit;
    }
    public static function req($k, $d = null) {
        if (isset($_POST[$k])) return $_POST[$k];
        if (isset($_GET[$k])) return $_GET[$k];
        return $d;
    }
    public static function clean($v) {
        if (is_string($v)) return trim($v);
        return $v;
    }
    public static function uploadImage($fileKey, $cfg) {
        if (!isset($_FILES[$fileKey])) return '';
        $f = $_FILES[$fileKey];
        if (empty($f['tmp_name'])) return '';
        if (!is_dir($cfg['upload_dir']))
mkdir($cfg['upload_dir'], 0755, true);

```

```

        $nm = preg_replace('/[^a-zA-Z0-9_\-\.\.]/', '_',
basename($f['name']));
        $dst = $cfg['upload_dir'] . '/' . uniqid('img_') . '_' .
$nm;

        move_uploaded_file($f['tmp_name'], $dst);
        return str_replace(__DIR__ . '/', '', $dst);
    }
    public static function now() {
        return date('Y-m-d H:i:s');
    }
    public static function int($v) {
        return intval($v);
    }
    public static function float($v) {
        return floatval($v);
    }
}

class ModelCategory {
    private $db;
    private $tbl = 'category';
    public function __construct($db) { $this->db = $db; }
    public function list() {
        return $this->db->fetchAll("SELECT * FROM `{ $this->tbl }`
ORDER BY `id` DESC");
    }
    public function get($id) {
        return $this->db->fetch("SELECT * FROM `{ $this->tbl }`
WHERE `id` = :id LIMIT 1", ['id' => $id]);
    }
    public function create($data) {
        $d = ['title' => Utils::clean($data['title'] ?? '')];
        return $this->db->insert($this->tbl, $d);
    }
    public function update($id, $data) {
        $d = ['title' => Utils::clean($data['title'] ?? '')];
        return $this->db->update($this->tbl, $d, 'id', $id);
    }
    public function delete($id) {
        return $this->db->delete($this->tbl, 'id', $id);
    }
}

class ModelCatalog {
    private $db;
    private $tbl = 'catalog';
    public function __construct($db) { $this->db = $db; }
    public function list($opts = []) {

```

```

        $sql = "SELECT * FROM `{ $this->tbl }`";
        $prs = [];
        $where = [];
        if (!empty($opts['category_id'])) {
            $where[] = "`category_id` = :cat";
            $prs['cat'] = $opts['category_id'];
        }
        if (!empty($opts['q'])) {
            $where[] = "(`name` LIKE :q OR `content` LIKE :q)";
            $prs['q'] = '%' . $opts['q'] . '%';
        }
        if (!empty($where)) $sql .= ' WHERE ' . implode(' AND ',
$where);
        $sql .= ' ORDER BY `id` DESC';
        return $this->db->fetchAll($sql, $prs);
    }
    public function get($id) {
        return $this->db->fetch("SELECT * FROM `{ $this->tbl }`
WHERE `id` = :id LIMIT 1", ['id' => $id]);
    }
    public function create($data) {
        $d = [];
        $d['name'] = Utils::clean($data['name'] ?? '');
        $d['content'] = Utils::clean($data['content'] ?? '');
        $d['category_id'] = Utils::int($data['category_id'] ??
0);
        $d['image'] = Utils::clean($data['image'] ?? '');
        $d['created_at'] = Utils::clean($data['created_at'] ??
Utils::now());
        $d['price'] = Utils::float($data['price'] ?? 0);
        $d['is_home'] = Utils::int($data['is_home'] ?? 0);
        $d['status'] = Utils::int($data['status'] ?? 1);
        return $this->db->insert($this->tbl, $d);
    }
    public function update($id, $data) {
        $d = [];
        if (isset($data['name'])) $d['name'] =
Utils::clean($data['name']);
        if (isset($data['content'])) $d['content'] =
Utils::clean($data['content']);
        if (isset($data['category_id'])) $d['category_id'] =
Utils::int($data['category_id']);
        if (isset($data['image'])) $d['image'] =
Utils::clean($data['image']);
        if (isset($data['price'])) $d['price'] =
Utils::float($data['price']);
        if (isset($data['is_home'])) $d['is_home'] =
Utils::int($data['is_home']);
    }

```

```

        if (isset($data['status'])) $d['status'] =
Utils::int($data['status']);
        if (empty($d)) return false;
        return $this->db->update($this->tbl, $d, 'id', $id);
    }
    public function delete($id) {
        return $this->db->delete($this->tbl, 'id', $id);
    }
}

class ModelComments {
    private $db;
    private $tbl = 'comments_catalog';
    public function __construct($db) { $this->db = $db; }
    public function listByCatalog($catalogId) {
        return $this->db->fetchAll("SELECT * FROM `{ $this->tbl }`
WHERE `catalog_id` = :cid ORDER BY `id` DESC", ['cid' =>
$catalogId]);
    }
    public function create($data) {
        $d = [];
        $d['name'] = Utils::clean($data['name'] ?? '');
        $d['content'] = Utils::clean($data['content'] ?? '');
        $d['catalog_id'] = Utils::int($data['catalog_id'] ?? 0);
        $d['created_at'] = Utils::clean($data['created_at'] ??
Utils::now());
        return $this->db->insert($this->tbl, $d);
    }
    public function delete($id) {
        return $this->db->delete($this->tbl, 'id', $id);
    }
}

class ModelOrders {
    private $db;
    private $tbl = 'orders';
    public function __construct($db) { $this->db = $db; }
    public function create($data) {
        $d = [];
        $d['name'] = Utils::clean($data['name'] ?? '');
        $d['deliver'] = Utils::clean($data['deliver'] ?? '');
        $d['city'] = Utils::clean($data['city'] ?? '');
        $d['address'] = Utils::clean($data['address'] ?? '');
        $d['catalog_id'] = Utils::int($data['catalog_id'] ?? 0);
        $d['create'] = Utils::clean($data['create'] ??
Utils::now());
        $d['note'] = Utils::clean($data['note'] ?? '');
        return $this->db->insert($this->tbl, $d);
    }
}

```

```

    }
    public function list() {
        return $this->db->fetchAll("SELECT * FROM `{ $this->tbl }`
ORDER BY `id` DESC");
    }
    public function get($id) {
        return $this->db->fetch("SELECT * FROM `{ $this->tbl }`
WHERE `id` = :id LIMIT 1", ['id' => $id]);
    }
    public function delete($id) {
        return $this->db->delete($this->tbl, 'id', $id);
    }
}

class Auth {
    private $db;
    public function __construct($db) { $this->db = $db; }
    public function login($u, $p) {
        $row = $this->db->fetch("SELECT * FROM `admins` WHERE
`username` = :u LIMIT 1", ['u' => $u]);
        if (!$row) return false;
        if (!$password_verify($p, $row['password'])) return false;
        $_SESSION['admin'] = ['id' => $row['id'], 'username' =>
$row['username']];
        return true;
    }
    public function logout() {
        unset($_SESSION['admin']);
        return true;
    }
    public function check() {
        return isset($_SESSION['admin']);
    }
}

class ControllerAPI {
    private $db;
    private $ctg;
    private $cat;
    private $cmt;
    private $ord;
    private $auth;
    private $cfg;
    public function __construct($db, $cfg) {
        $this->db = $db;
        $this->cfg = $cfg;
        $this->ctg = new ModelCategory($db);
        $this->cat = new ModelCatalog($db);
    }
}

```

```

        $this->cmt = new ModelComments($db);
        $this->ord = new ModelOrders($db);
        $this->auth = new Auth($db);
    }
    public function route() {
        $m = Utils::clean(Utils::req('m', 'site'));
        $a = Utils::clean(Utils::req('a', 'index'));
        if ($m === 'admin') $this->handleAdmin($a);
        else $this->handleSite($a);
    }
    private function handleAdmin($a) {
        if (!in_array($a, ['login', 'do_login']) && !$this->auth-
>check()) {
            Utils::json(['error' => 'unauth'], 401);
        }
        if ($a === 'login') {
            Utils::json(['ok' => true, 'msg' => 'ready']);
        }
        if ($a === 'do_login') {
            $u = Utils::clean(Utils::req('username', ''));
            $p = Utils::clean(Utils::req('password', ''));
            if ($this->auth->login($u, $p)) Utils::json(['ok' =>
true]);
            Utils::json(['error' => 'invalid'], 403);
        }
        if ($a === 'logout') {
            $this->auth->logout();
            Utils::json(['ok' => true]);
        }
        if ($a === 'categories_list') {
            $lst = $this->ctg->list();
            Utils::json(['data' => $lst]);
        }
        if ($a === 'categories_create') {
            $ttl = Utils::clean(Utils::req('title', ''));
            $id = $this->ctg->create(['title' => $ttl]);
            Utils::json(['ok' => true, 'id' => $id]);
        }
        if ($a === 'categories_update') {
            $id = Utils::int(Utils::req('id'));
            $ttl = Utils::clean(Utils::req('title', ''));
            $this->ctg->update($id, ['title' => $ttl]);
            Utils::json(['ok' => true]);
        }
        if ($a === 'categories_delete') {
            $id = Utils::int(Utils::req('id'));
            $this->ctg->delete($id);
            Utils::json(['ok' => true]);
        }
    }

```

```

    }
    if ($a === 'catalog_list') {
        $opts = [];
        if (Utils::req('category_id')) $opts['category_id'] =
Utils::int(Utils::req('category_id'));
        if (Utils::req('q')) $opts['q'] =
Utils::clean(Utils::req('q'));
        $lst = $this->cat->list($opts);
        Utils::json(['data' => $lst]);
    }
    if ($a === 'catalog_create') {
        $img = Utils::uploadImage('image', $this->cfg);
        $data = [
            'name' => Utils::clean(Utils::req('name','')),
            'content' =>
Utils::clean(Utils::req('content','')),
            'category_id' =>
Utils::int(Utils::req('category_id')),
            'image' => $img,
            'price' => Utils::float(Utils::req('price',0)),
            'is_home' => Utils::int(Utils::req('is_home',0)),
            'status' => Utils::int(Utils::req('status',1))
        ];
        $id = $this->cat->create($data);
        Utils::json(['ok' => true, 'id' => $id]);
    }
    if ($a === 'catalog_update') {
        $id = Utils::int(Utils::req('id'));
        $data = [];
        if (Utils::req('name') !== null) $data['name'] =
Utils::clean(Utils::req('name'));
        if (Utils::req('content') !== null) $data['content']
= Utils::clean(Utils::req('content'));
        if (Utils::req('category_id') !== null)
$data['category_id'] = Utils::int(Utils::req('category_id'));
        if (Utils::req('price') !== null) $data['price'] =
Utils::float(Utils::req('price'));
        if (isset($_FILES['image'])) {
            $img = Utils::uploadImage('image', $this->cfg);
            if ($img) $data['image'] = $img;
        }
        $this->cat->update($id, $data);
        Utils::json(['ok' => true]);
    }
    if ($a === 'catalog_delete') {
        $id = Utils::int(Utils::req('id'));
        $this->cat->delete($id);
        Utils::json(['ok' => true]);
    }

```

```

    }
    if ($a === 'orders_list') {
        $lst = $this->ord->list();
        Utils::json(['data' => $lst]);
    }
    if ($a === 'order_delete') {
        $id = Utils::int(Utils::req('id'));
        $this->ord->delete($id);
        Utils::json(['ok' => true]);
    }
    Utils::json(['error' => 'unknown_action'], 404);
}

private function handleSite($a) {
    if ($a === 'index') {
        $lst = $this->cat->list(['is_home' => 1]);
        Utils::json(['data' => $lst]);
    }
    if ($a === 'categories') {
        $lst = $this->ctg->list();
        Utils::json(['data' => $lst]);
    }
    if ($a === 'catalog') {
        $opts = [];
        if (Utils::req('category_id')) $opts['category_id'] =
Utils::int(Utils::req('category_id'));
        if (Utils::req('q')) $opts['q'] =
Utils::clean(Utils::req('q'));
        $lst = $this->cat->list($opts);
        Utils::json(['data' => $lst]);
    }
    if ($a === 'catalog_item') {
        $id = Utils::int(Utils::req('id'));
        $rw = $this->cat->get($id);
        if (!$rw) Utils::json(['error' => 'not_found'], 404);
        $cm = $this->cmt->listByCatalog($id);
        Utils::json(['item' => $rw, 'comments' => $cm]);
    }
    if ($a === 'comment_create') {
        $cid = Utils::int(Utils::req('catalog_id'));
        $d = [
            'name' => Utils::clean(Utils::req('name', '')),
            'content' =>
Utils::clean(Utils::req('content', '')),
            'catalog_id' => $cid,
            'created_at' => Utils::now()
        ];
        $id = $this->cmt->create($d);
        Utils::json(['ok' => true, 'id' => $id]);
    }
}

```

```

    }
    if ($a === 'order_create') {
        $cid = Utils::int(Utils::req('catalog_id'));
        $d = [
            'name' => Utils::clean(Utils::req('name','')),
            'deliver' =>
Utils::clean(Utils::req('deliver','')),
            'city' => Utils::clean(Utils::req('city','')),
            'address' =>
Utils::clean(Utils::req('address','')),
            'catalog_id' => $cid,
            'create' => Utils::now(),
            'note' => Utils::clean(Utils::req('note',''))
        ];
        $id = $this->ord->create($d);
        Utils::json(['ok' => true, 'id' => $id]);
    }
    Utils::json(['error' => 'unknown_action'], 404);
}
}

```

```

function db_crt_conn() {
    $hst = "localhost";
    $usr = "rtdb";
    $pss = "psswd";
    $dbn = "ntqsys";
    $cnn = mysqli_connect($hst, $usr, $pss, $dbn);
    return $cnn;
}

```

```

function db_cls_conn($cnn) {
    mysqli_close($cnn);
}

```

```

function db_slct_all($tbl) {
    $cnn = db_crt_conn();
    $qr = "SELECT * FROM $tbl";
    $rs = mysqli_query($cnn, $qr);
    $dt = [];
    while($rw = mysqli_fetch_assoc($rs)) {
        $dt[] = $rw;
    }
    db_cls_conn($cnn);
    return $dt;
}

```

```

function db_slct_id($tbl, $id) {

```

```

    $cnn = db_crt_conn();
    $qrr = "SELECT * FROM $tbl WHERE id='$id'";
    $rs = mysqli_query($cnn, $qrr);
    $rw = mysqli_fetch_assoc($rs);
    db_cls_conn($cnn);
    return $rw;
}

function db_ins_itm($nm, $prc, $stt, $ctg) {
    $cnn = db_crt_conn();
    $qrr = "INSERT INTO tms (nm, prc, stt, ctg) VALUES ('$nm',
'$prc', '$stt', '$ctg')";
    mysqli_query($cnn, $qrr);
    db_cls_conn($cnn);
}

function db_upd_itm($id, $nm, $prc, $stt, $ctg) {
    $cnn = db_crt_conn();
    $qrr = "UPDATE tms SET nm='$nm', prc='$prc', stt='$stt',
ctg='$ctg' WHERE id='$id'";
    mysqli_query($cnn, $qrr);
    db_cls_conn($cnn);
}

function db_dlt_itm($id) {
    $cnn = db_crt_conn();
    $qrr = "DELETE FROM tms WHERE id='$id'";
    mysqli_query($cnn, $qrr);
    db_cls_conn($cnn);
}

function sl_itm($id, $qnt) {
    $cnn = db_crt_conn();
    $rw = db_slct_id("tms", $id);
    $prc = $rw['prc'];
    $ttl = $qnt * $prc;
    $dt = date("Y-m-d H:i:s");
    $qrr = "INSERT INTO sls (tm_id, qnt, ttl, dt) VALUES
('$id', '$qnt', '$ttl', '$dt')";
    mysqli_query($cnn, $qrr);
    db_cls_conn($cnn);
}

function sl_gn_rcpt($sl_id) {
    $cnn = db_crt_conn();
    $qrr = "SELECT * FROM sls WHERE id='$sl_id'";
    $rs = mysqli_query($cnn, $qrr);
    $rw = mysqli_fetch_assoc($rs);

```

```

    $tm = db_slct_id("tms", $rw['tm_id']);
    $rcpt = "";
    $rcpt .= "RCPT - " . $sl_id . "\n";
    $rcpt .= "TM: " . $tm['nm'] . "\n";
    $rcpt .= "QNT: " . $rw['qnt'] . "\n";
    $rcpt .= "TTL: " . $rw['ttl'] . "\n";
    $rcpt .= "DT: " . $rw['dt'];
    db_cls_conn($cnn);
    return $rcpt;
}

function rpdt_dly_sls() {
    $cnn = db_crt_conn();
    $dy = date("Y-m-d");
    $qr = "SELECT * FROM sls WHERE dt LIKE '$dy%'";
    $rs = mysqli_query($cnn, $qr);
    $dt = [];
    while($rw = mysqli_fetch_assoc($rs)) {
        $dt[] = $rw;
    }
    db_cls_conn($cnn);
    return $dt;
}

function rpdt_mnth_sls() {
    $cnn = db_crt_conn();
    $mn = date("Y-m");
    $qr = "SELECT * FROM sls WHERE dt LIKE '$mn%'";
    $rs = mysqli_query($cnn, $qr);
    $dt = [];
    while($rw = mysqli_fetch_assoc($rs)) {
        $dt[] = $rw;
    }
    db_cls_conn($cnn);
    return $dt;
}

function rpdt_top_tm() {
    $cnn = db_crt_conn();
    $qr = "SELECT tm_id, SUM(qnt) AS sm FROM sls GROUP BY tm_id
ORDER BY sm DESC LIMIT 1";
    $rs = mysqli_query($cnn, $qr);
    $rw = mysqli_fetch_assoc($rs);
    $tm = db_slct_id("tms", $rw['tm_id']);
    db_cls_conn($cnn);
    return $tm;
}

```

```

function usr_rgstr($lg, $ps) {
    $cnn = db_crt_conn();
    $psh = md5($ps);
    $qr = "INSERT INTO usr (lg, ps) VALUES ('$lg','$psh')";
    mysqli_query($cnn, $qr);
    db_cls_conn($cnn);
}

function usr_lgn($lg, $ps) {
    $cnn = db_crt_conn();
    $psh = md5($ps);
    $qr = "SELECT * FROM usr WHERE lg='$lg' AND ps='$psh'";
    $rs = mysqli_query($cnn, $qr);
    $cnt = mysqli_num_rows($rs);
    db_cls_conn($cnn);
    if($cnt > 0) {
        $_SESSION['usr'] = $lg;
        return true;
    } else {
        return false;
    }
}

function usr_lgt() {
    session_start();
    session_destroy();
}

function inv_chck_stk($id) {
    $rw = db_slct_id("tms", $id);
    if($rw['stt'] == "sl") {
        return false;
    }
    return true;
}

function inv_mrk_sl($id) {
    $cnn = db_crt_conn();
    $qr = "UPDATE tms SET stt='sl' WHERE id='$id'";
    mysqli_query($cnn, $qr);
    db_cls_conn($cnn);
}

function ntf_nt_sl() {
    $cnn = db_crt_conn();
    $qr = "SELECT * FROM tms WHERE stt='sl'";
    $rs = mysqli_query($cnn, $qr);
    $cnt = mysqli_num_rows($rs);
}

```

```
db_cls_conn($cnn);  
if($cnt > 0) {  
    return "sl ntf";  
}  
return "ns ntf";  
}
```

ДОДАТОК В
Слайди презентації

Створення програмної системи опрацювання даних для продажу антикваріату

Виконав	Денис ПИСАНЕЦЬ ст.гр. КНТ-222
Керівник	Андрій ОЛІЙНИК д.т.н., професор

Рисунок В.1 – Слайд 1

Об'єкт, предмет та мета роботи

Об'єкт роботи – процеси обчислень, пов'язані з організацією та опрацюванням даних для продажу антикваріату.

Предмет роботи – програмні системи опрацювання даних для продажу антикваріату.

Мета роботи – розробка програмної системи опрацювання даних для продажу антикваріату для підвищення ефективності розповсюдження товарів.

Рисунок В.2 – Слайд 2

Задачі роботи

- виконати аналіз предметної області та систем опрацювання даних для продажу антикваріату;
- здійснити проектування системи опрацювання даних для продажу антикваріату;
- створити систему опрацювання даних для продажу антикваріату;
- дослідити розроблену систему опрацювання даних для продажу антикваріату.

Рисунок В.3 – Слайд 3

Порівняння існуючих аналогів

Критерій порівняння	Cashier Live	Simple Consign	Go Antiquing
Хмарна архітектура	+	+	+—
Інструменти роботи з постачальниками	+—	+	+
Розширені можливості управління консигнаціями	+	+—	+—
Підтримка багатьох локацій (мультивітринність)	+	+—	+—
Інтеграція з бухгалтерськими системами	—	+	—
Невисока вартість використання	+—	+—	+

Рисунок В.4 – Слайд 4

Вибір мови програмування

Критерій порівняння мов програмування	Мова програмування		
	PHP	JavaScript	Ruby
Швидкість розроблення	+	+-	+-
Зручність для вебплатформ	+	+	+-
Простота інтеграції з базами даних	+	+-	-
Масштабованість	+	+-	+
Наявність готових бібліотек і модулів	+	+	+-

Рисунок В.5 – Слайд 5

Вибір середовища розробки

Критерій порівняння середовищ розробки	Середовища розробки		
	Visual Studio Code	PhpStorm	Aptana Studio
Можливість гнучкого розширення	+	+-	+-
Підтримка плагінів для баз даних	+	+	+-
Інструменти налагодження і тестування	+	+	-
Підтримка роботи з вебфреймворками	+	+	+-
Низькі вимоги до обладнання	+	-	+-

Рисунок В.6 – Слайд 6

Вибір засобів зберігання та опрацювання даних

Критерій порівняння систем управління базами даних	Системи управління базами даних		
	MySQL	PostgreSQL	Oracle Database
Швидкість обробки запитів	+	+—	+—
Простота налаштування та розгортання	+	+—	—
Інтеграція з вебтехнологіями	+	+	+—
Невисока вартість використання	+	+	+—
Простота резервного копіювання	+	+	+—
Активність спільноти	+	+—	—

Рисунок В.7 – Слайд 7

Структура системи опрацювання даних для продажу антикваріату

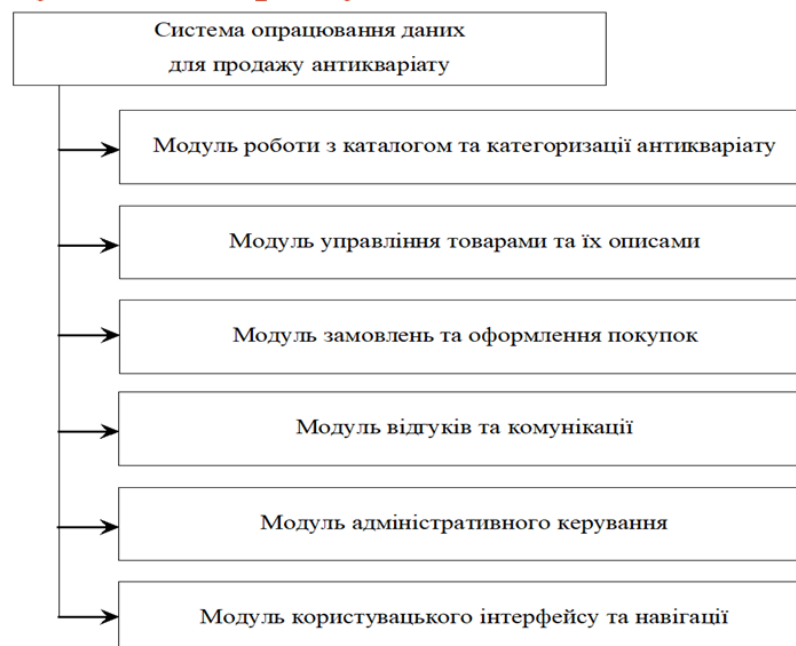


Рисунок В.8 – Слайд 8

Схема функціонування програми

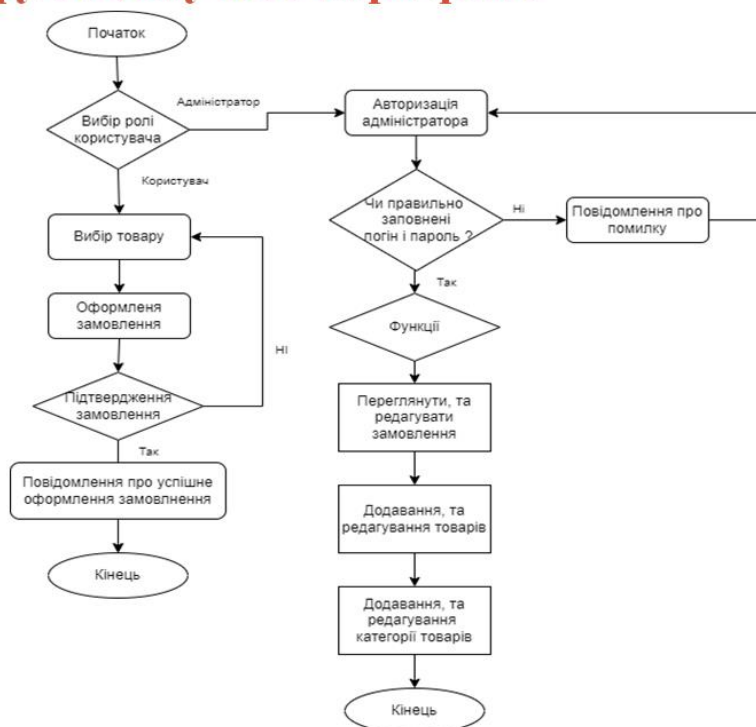


Рисунок В.9 – Слайд 9

Схема бази даних

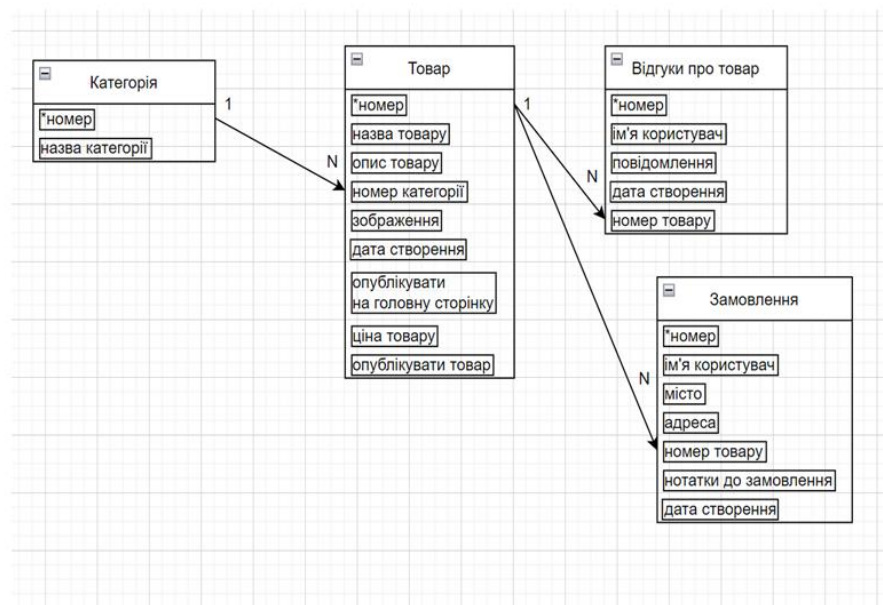


Рисунок В.10 – Слайд 10

Робота з програмою. Головна сторінка. Сторінка антикварного товару

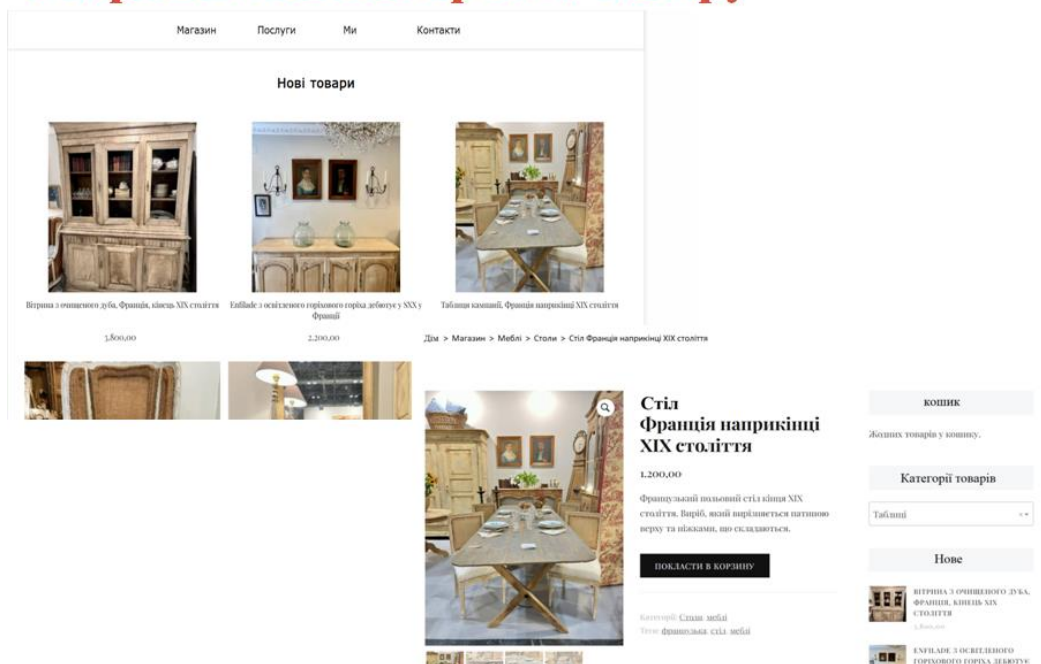


Рисунок В.11 – Слайд 11

Робота з програмою (кошик, замовлення)

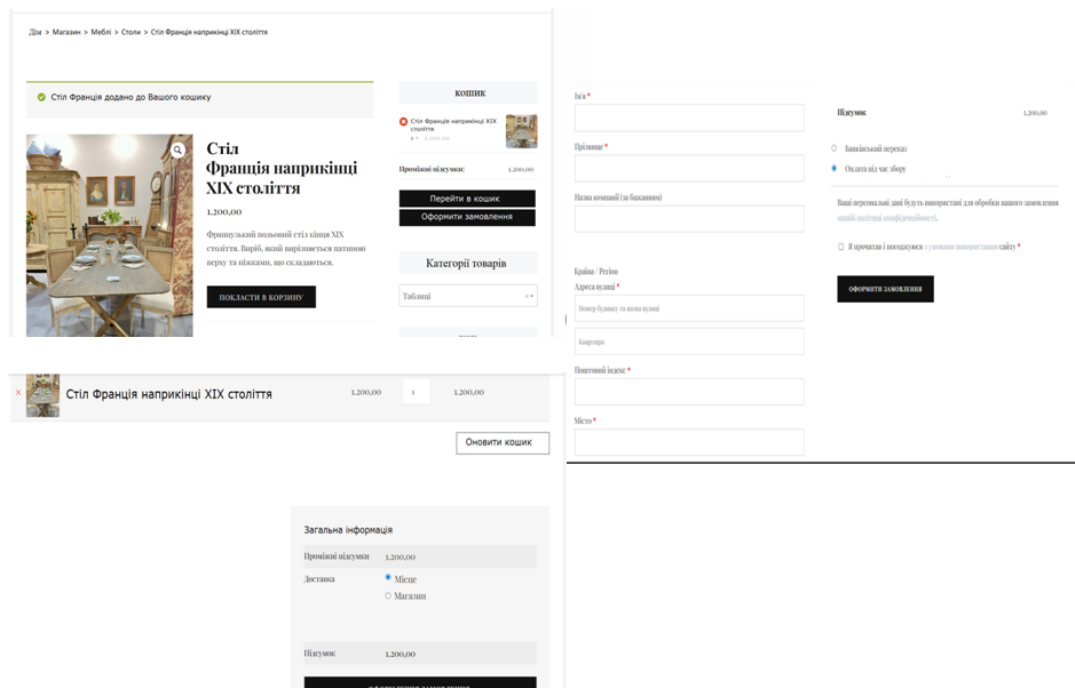


Рисунок В.12 – Слайд 12

Висновки

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процеси обчислень, пов'язані з організацією та опрацюванням даних для продажу антикваріату.

Сформульовано функціональні вимоги до системи опрацювання даних для продажу антикваріату. Для створення системи опрацювання даних для продажу антикваріату обрано мову програмування PHP та середовище розробки Visual Studio Code. Для зберігання та опрацювання даних обрано систему управління базами даних MySQL.

Розроблено програмне систему опрацювання даних для продажу антикваріату. Розроблено базу даних системи опрацювання даних для продажу антикваріату, яка побудована таким чином, щоб забезпечувати повне і структуроване зберігання інформації про товари, замовлення, категорії та відгуки покупців. Виконано тестування розробленої системи опрацювання даних для продажу антикваріату.

Усі завдання дипломної кваліфікаційної роботи бакалавра повністю виконано.

Рисунок В.13 – Слайд 13