

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра інженерії програмного забезпечення



## КВАЛІФІКАЦІЙНА РОБОТА

Вебсайт онлайн-галереї з продажу картин

Назва теми

Рівень вищої освіти Перший(бакалаврський)  
Галузь знань 12 «Інформаційні технології»  
Спеціальність 121 «Інженерія програмного забезпечення»  
Освітня програма Освітньо-професійна програма «Інженерія програмного  
забезпечення»

Шифр КвРІПЗ. 220196.01.03.ПЗ

Виконав студент IV курсу, група ІПЗ-22-1  Максим БОЖИК  
Підпис Ім'я, ПРІЗВИЩЕ  
Керівник канд. техн. наук, доцент  Юрій ФОРКУН  
Науковий ступінь, вчене звання Підпис Ім'я, ПРІЗВИЩЕ  
Нормоконтроль канд. техн. наук, доцент  Оксана ЯШИНА  
Підпис Ім'я, ПРІЗВИЩЕ

До захисту допускаю:

Завідувач кафедри інженерії  
програмного забезпечення

  
Підпис

Леонід БЕДРАТЮК  
Ім'я, ПРІЗВИЩЕ

8 червня 2026 р.

Хмельницький 2026

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Інженерії програмного забезпечення

Рівень вищої освіти Перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри 193

Л. П. Бедратюк

07. 01 2026 р.

**ЗАВДАННЯ**

**НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Божик Максим Сергійович

Прізвище, ім'я, по батькові студента

1. Тема кваліфікаційної роботи Вебсайт онлайн-галереї з продажу картин

Керівник кваліфікаційної роботи Форкун Юрій Вікторович, канд.техн наук, доцент

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

Затверджена наказом ректора університету від 20.01.2026 р. №7

2. Строк подання студентом роботи на кафедру 01.06.2026 р.

3. Вихідні дані до роботи Матеріали переддипломної практики

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

Дослідження предметної області та постановка задачі, аналіз існуючих рішень, проєктування застосунку, програмна реалізація та тестування

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

Презентаційні матеріали (слайди), діаграми варіантів використання, діаграми архітектурних рішень, ER-діаграма.

6. Консультанти розділів кваліфікаційної роботи

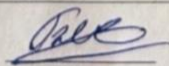
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Яшина О.М., доцент	05.05.26	20.05.26
Антиплагіат	Форкун Ю.В., доцент	05.05.26	25.05.26

7. Дата видачі завдання « 02 » 01 2026 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1 Ознайомлення з тематикою дипломного проєктування, визначення та узгодження індивідуальної теми кваліфікаційних робіт (КвР)	01.12 – 31.12.2025	
2 Збір матеріалу за темою КвР; дослідження предметної області, в якій планується використання програмного забезпечення (ПЗ), визначення задач та вимог, розробка технічного завдання	01.01 – 20.02.2026	
3 Проєктування програмного забезпечення	21.02 – 20.03.2026	
4 Програмна реалізація з використанням відповідних засобів розробки. Тестування ПЗ	21.03 – 30.04.2026	
5 Написання вступу, загальних висновків, оформлення переліку джерел посилання та додатків. Оформлення пояснювальної записки КвР згідно вимог	01.05 – 25.05.2026	
6 Попередній захист КвР	Травень 2026	
7 Перевірка КвР на плагіат, нормоконтроль, отримання відгуків, рецензій та інших супровідних документів. Брошування (зшиття) пояснювальної записки.	26.05 – 30.05.2026	
8 Здача КвР на кафедру; підготовка КвР для розміщення у репозитарії ХНУ; підготовка до захисту та захист КвР	з 01.06.2026	

Студент

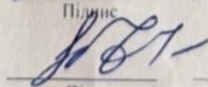


Підпис

М. С. Божик

Ініціали, прізвище

Керівник роботи



Підпис

Ю. В. Форкун

Ініціали, прізвище

## АНОТАЦІЯ

Тема бакалаврської кваліфікаційної роботи: «Вебсайт онлайн-галереї з продажу картин»

Автор роботи: Божик Максим Сергійович

Керівник роботи: Форкун Юрій Вікторович.

Пояснювальна записка: 82 с., 20 рис., 15 табл., 4 дод., 48 джерела.

ВЕБСАЙТ, ІНТЕРНЕТ-МАГАЗИН, ЕЛЕКТРОННА КОМЕРЦІЯ, ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАСТОСУНКУ, АУТЕНТИФІКАЦІЯ, ПЕРЕГЛЯД КАРТИН, ФІЛЬТРАЦІЯ, КОШИК ЗАМОВЛЕНЬ.

Метою кваліфікаційної роботи є розробка вебсайту онлайн-галереї з продажу картин.

У ході роботи над даним програмним застосунком було проаналізовано предметну область задачі, яка стосується продажу картин, переглянуто існуючі рішення, сформульовано мету кваліфікаційної роботи та виділено ряд підзадач, для реалізувати для досягнення поставленої мети роботи.

Як платформу розробки використано ASP.NET Core MVC, середовище програмування Microsoft Visual Studio, систему керування базами даних Microsoft SQL Server та технологію Entity Framework Core. Для створення клієнтської частини використано HTML5, CSS3, JavaScript та Bootstrap.

У результаті проведеної роботи було розроблено вебсайт онлайн-галереї з продажу картин, що надасть змогу користувачам знайти, що їх цікавлять, комунікувати один з одним, залишати відгуки та інформаційні повідомлення тощо.

26.05.2026

ЗМІСТ ..... 4

ВСТУП..... 6

1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ... 10

1.1 Змістовий аналіз предметної області, її структурних та функціональних особливостей..... 10

1.2 Аналіз наявного програмно-технічного забезпечення предметної області ..... 15

1.3 Визначення функціональних та нефункціональних вимог до програмного забезпечення ..... 21

1.4 Висновки. Постановка задачі..... 28

2. ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ..... 30

2.1 Вибір типу архітектури та шаблонів проєктування..... 30

2.2 Детальне проєктування програмного забезпечення ..... 34

2.3 Опис залежностей ..... 42

2.4 Висновки другого розділу ..... 51

3. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ..... 53

3.1 Створення бази даних проєкту ..... 53

3.2 Реалізація серверної та клієнтської частин програмного забезпечення ..... 59

3.3 Реалізація клієнтської частини програмного забезпечення та інтерфейсу 67

КвРПЗ.220196.01.03.ПЗ

Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Божик М.С.		28.05	Літ.	Арк.	Акрушів
Перевір.		Форкун Ю.В.		28.05		7	
Реценз.					ХНУ. ІПЗ-22-1		
Н. Контр.		Ашчина О.М.		25.05			
Затверд.		Бедратюк Л.П.					

Вебсайт онлайн-галереї з продажу картин

3.5 Тестування програмного забезпечення.....	68
3.6. Висновки третього розділу.....	71
ВИСНОВКИ.....	75
ДЖЕРЕЛА.....	78
ДОДАТОК А.....	83
ДОДАТОК Б.....	87
ДОДАТОК В.....	91
ДОДАТОК В.....	111

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						6
Змін.	Арк.	№ докум.	Підпис.	Дата		

## ВСТУП

У сучасних умовах стрімкого та швидкого розвитку інформаційних технологій та цифровізації суспільства та світу особливого значення набувають веборієнтовані системи для електронної комерції. Використання ресурсів сучасної мережі Інтернет як для пошуку, демонстрації та продажу товарів стало вже невід'ємною складовою діяльності багатьох підприємств, а також творчих організацій. Одним із таких сучасних та перспективних напрямів електронної комерції є реалізація та продаж творів мистецтва через спеціалізовані вебресурси, які забезпечують досить зручну взаємодію між митцями, художниками, галереями та потенційними покупцями творів мистецтва.

Онлайн-галереї картин дозволяють усім користувачам сайтів переглядати мистецькі твори незалежно від свого місця перебування, отримувати детальну інформацію про авторів та їхні мистецькі твори, а також здійснювати покупки уподобаного в декілька кліків. Водночас власники таких вебплатформ отримують ефективний інструмент для свого управління каталогом товарів, обробки замовлень, контролю продажів творів мистецтва та підтримки актуальності представленої про них інформації. Саме тому розробка таких сучасних вебсайтів для продажу картин та інших творів мистецтва є актуальним завданням у сфері інженерії програмного забезпечення.

Об'єктом дослідження тут є процеси функціонування онлайн-галереї з продажу картин та допоміжні програмні системи, що забезпечують взаємодію між користувачами, сайтом та адміністрацією вебресурсу.

Предметом дослідження виступають методи, засоби та технології проектування і розробки вебзастосунків для електронної комерції, а також різні програмні рішення для створення онлайн-галереї картин та творів мистецтва.

Метою нашої кваліфікаційної роботи є проектування архітектури та розробка вебсайту онлайн-галереї з продажу картин, який забезпечує зручний перегляд

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						7
Змін.	Арк.	№ докум.	Підпис.	Дата		

художніх мистецьких творів, пошук необхідної інформації про картини, оформлення замовлень та ефективно і зручне адміністрування контенту.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз предметної області та дослідити особливості функціонування сучасних онлайн-галерей і вебмагазинів творів мистецтва.
- визначити функціональні та нефункціональні вимоги до вебсайту.
- розробити структуру та навігаційну модель вебресурсу.
- спроектувати архітектуру програмного забезпечення та структуру бази даних.
- розробити користувацький вебінтерфейс із забезпеченням зручності використання та адаптивності для різних пристроїв.
- реалізувати механізми перегляду каталогу картин, пошуку, фільтрації та оформлення замовлень.
- створити адміністративну панель для управління картинами, категоріями, користувачами та замовленнями.
- забезпечити безпечне зберігання та обробку даних користувачів.
- провести тестування розробленої системи та оцінити її працездатність.
- виконати аналіз результатів розробки та визначити можливі напрями подальшого вдосконалення вебресурсу.

Під час проектування та розробки вебсайту нами було поставлено загальні задачі створення надійної, масштабованої та зручної інформаційної системи, яка забезпечує ефективну взаємодію між продавцем і покупцем, підтримує актуальність каталогу картин, дозволяє автоматизувати процес приймання та обробки замовлень, а також забезпечує високий рівень продуктивності та зручності користування.

Практичне значення роботи полягає у створенні повноцінного вебсайту онлайн-галереї, який може використовуватися художниками, мистецькими студіями, галереями та іншими організаціями для демонстрації та реалізації творів мистецтва через мережу Інтернет. Розроблений програмний продукт сприяє

					КвРПЗ.220196.01.03.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис.	Дата		8

розширенню аудиторії потенційних покупців, підвищенню ефективності продажів та автоматизації бізнес-процесів.

Для реалізації програмного забезпечення нами використовувалося середовище розробки Visual Studio Code, яке є сучасним редактором вихідного коду з широкими можливостями налаштування та підтримкою великої кількості мов програмування, фреймворків і розширень. Використання даного інструменту дозволило ефективно організувати процес розробки, тестування та супроводу нашого вебзастосунку.

					КвРІПЗ.220196.01.03.ІЗ	Арк.
						9
Змін.	Арк.	№ докум.	Підпис.	Дата		

# 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

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

У сучасних умовах стрімкого розвитку інформаційних технологій та електронної комерції все більшої популярності набувають онлайн-платформи, призначені для продажу товарів та послуг через мережу Інтернет. Одним із перспективних напрямків розвитку електронної торгівлі є реалізація творів мистецтва, зокрема картин, через спеціалізовані вебресурси. Використання сучасних вебтехнологій дозволяє значно спростити процес взаємодії між художниками, продавцями та потенційними покупцями, забезпечуючи швидкий доступ до інформації про мистецькі твори, можливість їх придбання та зручне управління замовленнями.

Предметною областю даного дослідження є процес організації онлайн-продажу картин за допомогою вебсайту галерейного типу. Такий вебсайт виступає посередником між автором або власником художнього твору та кінцевим покупцем, забезпечуючи повний цикл продажу: від ознайомлення користувача з асортиментом до оформлення та оплати замовлення.

Останніми роками ринок електронної комерції демонструє стабільне зростання. Збільшення кількості користувачів мережі Інтернет, поширення мобільних пристроїв та вдосконалення платіжних систем сприяють тому, що дедалі більше покупців надають перевагу онлайн-покупкам. Ця тенденція поширюється не лише на товари повсякденного вжитку, але й на предмети мистецтва. Покупці отримують можливість переглядати роботи художників з різних регіонів світу, порівнювати їх характеристики, знайомитися з історією створення картин та здійснювати покупки без необхідності особистого відвідування художніх галерей.

Важливою особливістю сучасних вебсайтів електронної комерції є використання інтелектуальних механізмів пошуку, рекомендацій та персоналізації. Дані технології дозволяють підвищити якість взаємодії користувача із системою та

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						10
Змін.	Арк.	№ докум.	Підпис.	Дата		

забезпечити індивідуальний підхід до кожного відвідувача. Пошукові механізми допомагають швидко знаходити необхідні товари, системи рекомендацій пропонують користувачам картини відповідно до їх інтересів, а персоналізація дозволяє адаптувати вміст сайту до потреб конкретного покупця.

Пошук є одним із ключових елементів будь-якого інтернет-магазину. Саме з нього найчастіше починається взаємодія користувача із системою. Від ефективності роботи пошукового механізму залежить швидкість знаходження необхідного товару та загальне враження користувача від вебресурсу. Сучасні системи пошуку використовують алгоритми машинного навчання, технології обробки природної мови та аналіз поведінки користувачів для формування максимально релевантних результатів.

Система рекомендацій є ще одним важливим компонентом електронної комерції. Її основне завдання полягає у пропонуванні користувачам товарів, які можуть їх зацікавити на основі попередньої поведінки, переглядів або покупок. Для вебсайту продажу картин рекомендаційна система може враховувати улюблені стилі живопису користувача, кольорову гаму, тематику робіт або популярність конкретних художників.

Персоналізація вебсайту забезпечує адаптацію інтерфейсу та контенту відповідно до потреб кожного користувача. Завдяки персоналізації покупець може отримувати індивідуальні пропозиції, рекомендації та повідомлення, що сприяє підвищенню рівня задоволеності від використання ресурсу та збільшенню кількості успішних продажів.

Основною метою функціонування вебсайту онлайн-галереї є забезпечення ефективного процесу продажу картин через мережу Інтернет. Для досягнення цієї мети система повинна реалізовувати ряд взаємопов'язаних функцій, спрямованих на підтримку діяльності як покупців, так і адміністрації ресурсу.

До основних учасників нашої предметної області належать:

- адміністратор системи;
- зареєстрований користувач сайту;
- незареєстрований відвідувач сайту;

					КвРІПЗ.220196.01.03.ІІЗ	Арк.
Змін.	Арк.	№ докум.	Підпис.	Дата		11

- художник або постачальник картин;
- платіжна система;
- способи та служба доставки.

Кожен із перелічених учасників взаємодіє з вебсайтом відповідно до своїх функціональних обов'язків.

Адміністратор системи здійснює управління контентом вебсайту, контролює додавання нових картин, редагує інформацію про товари, переглядає замовлення, керує категоріями та забезпечує коректне функціонування всієї системи.

Незарєєстрований користувач сайту має можливість переглядати каталог картин, використовувати систему пошуку, знайомитися з інформацією про художників та окремі роботи. Для оформлення замовлення користувачеві необхідно пройти процедуру реєстрації або авторизації.

Зарєєстрований користувач сайту отримує розширений функціонал, який включає можливість формування кошика покупок, оформлення замовлень, перегляду історії покупок, редагування особистих даних та збереження обраних картин до списку бажань.

Художник або ж постачальник забезпечує наповнення каталогу новими роботами, надає інформацію про картини та контролює наявність доступних для продажу екземплярів.

Платіжна система сайту відповідає за проведення фінансових операцій та забезпечує безпечне виконання онлайн-платежів.

Способи та служба доставки здійснює транспортування придбаних картин до кінцевого покупця відповідно до зазначеної адреси користувачем.

Структура предметної області включає декілька взаємопов'язаних компонентів. Центральним елементом виступає каталог картин, який містить повну інформацію про всі товари, доступні для продажу. Кожна картина характеризується певним набором параметрів, серед яких можна назвати:

- назва картини;
- ім'я автора картини;

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						12
Змін.	Арк.	№ докум.	Підпис.	Дата		

- стиль виконання картини;
- жанр картини;
- техніка створення картини;
- матеріал основи картини;
- розміри картини;
- рік створення картини;
- ціна картини;
- наявність у продажу;
- короткий та розгорнутий опис картини;
- фотографії у високій якості.

Наявність детальної інформації про товар є надзвичайно важливою особливістю онлайн-продажу предметів мистецтва, оскільки покупець не має можливості безпосередньо оглянути картину перед придбанням.

Каталог повинен підтримувати структурування картин за категоріями. Це забезпечує швидкий пошук необхідних робіт та покращує навігацію по сайту. Основними категоріями картин на нашу думку можуть бути:

- пейзажі;
- натюрморти;
- портрети;
- абстракція;
- сучасне мистецтво;
- анімалістика;
- морська тематика;
- архітектурний живопис;
- історичний жанр;
- декоративний живопис.

Для зручності користувачів наш вебсайт повинен підтримувати механізми фільтрації товарів за різними критеріями. До таких критеріїв належать ціна,

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						13
Змін.	Арк.	№ докум.	Підпис.	Дата		

художник, стиль, розмір картини, матеріал виконання, дата створення та популярність.

Однією з найважливіших складових предметної області є процес оформлення певного замовлення. Він включає декілька етапів:

1. Вибір картини.
2. Додавання товару до кошика.
3. Перевірка складу замовлення.
4. Введення контактних даних.
5. Вибір способу доставки.
6. Вибір способу оплати.
7. Підтвердження замовлення.
8. Виконання платежу.
9. Обробка замовлення адміністратором.
10. Передача товару службі доставки.

Автоматизація зазначених процесів дозволяє мінімізувати кількість помилок та підвищити ефективність роботи інтернет-магазину.

Особливе значення для вебсайту продажу картин має якісний користувацький інтерфейс. Дизайн ресурсу повинен відповідати специфіці художньої тематики та створювати позитивне емоційне враження у відвідувачів. Важливо забезпечити гармонійне поєднання кольорової гами, зручної навігації та високої швидкодії вебресурсу.

Під час розроблення сучасних вебсайтів широко використовуються вебресурси, що забезпечують функціонування клієнтської частини системи. До них належать HTML-документи, таблиці стилів CSS, сценарії JavaScript, графічні файли та мультимедійний контент. Саме ці компоненти формують інтерфейс користувача та забезпечують інтерактивність вебдодатку.

HTML використовується для створення структури вебсторінок та визначення їх змісту. За допомогою CSS реалізується оформлення елементів інтерфейсу та адаптація дизайну під різні типи пристроїв. JavaScript забезпечує динамічну

					КвРІПЗ.220196.01.03.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис.	Дата		14

поведінку сторінок, взаємодію з сервером без перезавантаження сторінки та реалізацію сучасних інтерактивних можливостей.

У зв'язку зі значним поширенням мобільних пристроїв важливим аспектом функціонування вебсайту є адаптивність інтерфейсу. Система повинна коректно відображатися на персональних комп'ютерах, планшетах та смартфонах незалежно від роздільної здатності екрана.

Окрему увагу необхідно приділяти питанням інформаційної безпеки. Вебсайт повинен забезпечувати захист персональних даних користувачів, безпечне зберігання паролів, шифрування мережевого трафіку та захист від несанкціонованого доступу. Для цього використовуються сучасні механізми автентифікації, авторизації та криптографічного захисту інформації з урахуванням всіх можливих потенційних загроз.

Значну роль у роботі онлайн-галереї відіграє база даних. Саме вона забезпечує централізоване зберігання інформації про користувачів, картини, категорії, замовлення та платежі. Використання систем керування базами даних дозволяє ефективно обробляти великі обсяги інформації та забезпечувати її цілісність та узгодженість.

Таким чином, предметна область вебсайту онлайн-галереї з продажу картин характеризується складною структурою взаємопов'язаних об'єктів та бізнес-процесів. Її основними компонентами є каталог картин, система керування користувачами, механізми пошуку та фільтрації, кошик покупок, система оформлення замовлень, модулі оплати та доставки. Автоматизація зазначених процесів за допомогою сучасних вебтехнологій дозволяє підвищити ефективність продажу творів мистецтва, забезпечити зручність користування ресурсом та розширити аудиторію потенційних покупців.

## 1.2 Аналіз наявного програмно-технічного забезпечення предметної області

Перед розробкою вебсайту онлайн-галереї з продажу картин доцільно провести аналіз існуючих програмних рішень, представлених на ринку електронної

					КвРІПЗ.220196.01.03.ІЗ	Арк.
Змін.	Арк.	№ докум.	Підпис.	Дата		15

комерції у сфері реалізації мистецьких творів. Такий аналіз дозволяє визначити переваги та недоліки вже існуючих вебресурсів, виявити найбільш ефективні підходи до організації інтерфейсу користувача та сформулювати перелік функціональних можливостей, які доцільно реалізувати у нашому програмному продукті, а саме веб-сайті.

На сьогоднішній день існує досить велика кількість вебсайтів, що спеціалізуються на продажі картин, художніх репродукцій, постерів та інших декоративних елементів інтер'єру. Більшість таких ресурсів функціонують як інтернет-магазини або онлайн-галереї, що забезпечують перегляд каталогу продукції, пошук товарів, оформлення замовлень та проведення електронних платежів тощо.

Для аналізу нами були обрані три популярні ресурси, які здійснюють продаж картин через мережу Інтернет, а саме:

- ArtSale.ua;
- Art-Holst;
- Jose Art Gallery.

Вибір нами саме цих вебресурсів обумовлений їх популярністю серед користувачів, різними підходами до організації інтерфейсу та наявністю функціоналу, характерного для сучасних систем електронної комерції.

На рисунку 1.1 наведено головну сторінку вебресурсу ArtSale.ua.

Проаналізувавши структуру сайту ArtSale.ua, можна зробити висновок, що ресурс орієнтований на швидкий доступ користувача до каталогу продукції. У верхній частині сторінки цього вебсайту розташовані основні елементи навігації, серед яких каталог картин, контакти, інформація про оплату та доставку, акційні пропозиції та блог .

Важливою перевагою даного ресурсу є наявність пошукового рядка, який знаходиться у центральній частині верхньої панелі. Це дозволяє користувачеві швидко знайти необхідний товар без тривалого перегляду каталогу. Також реалізовано багаторівневе меню категорій, що забезпечує зручну навігацію між різними видами картин.

					КвРІПЗ.220196.01.03.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис.	Дата		16





До недоліків можна віднести обмежену кількість функціональних елементів на головній сторінці та відсутність додаткових сервісів взаємодії з користувачами, які присутні у деяких конкурентних рішеннях.

Наступним ресурсом для аналізу нами було обрано міжнародну онлайн-галерею Jose Art Gallery, головна сторінка якої представлена на рисунку 1.3.

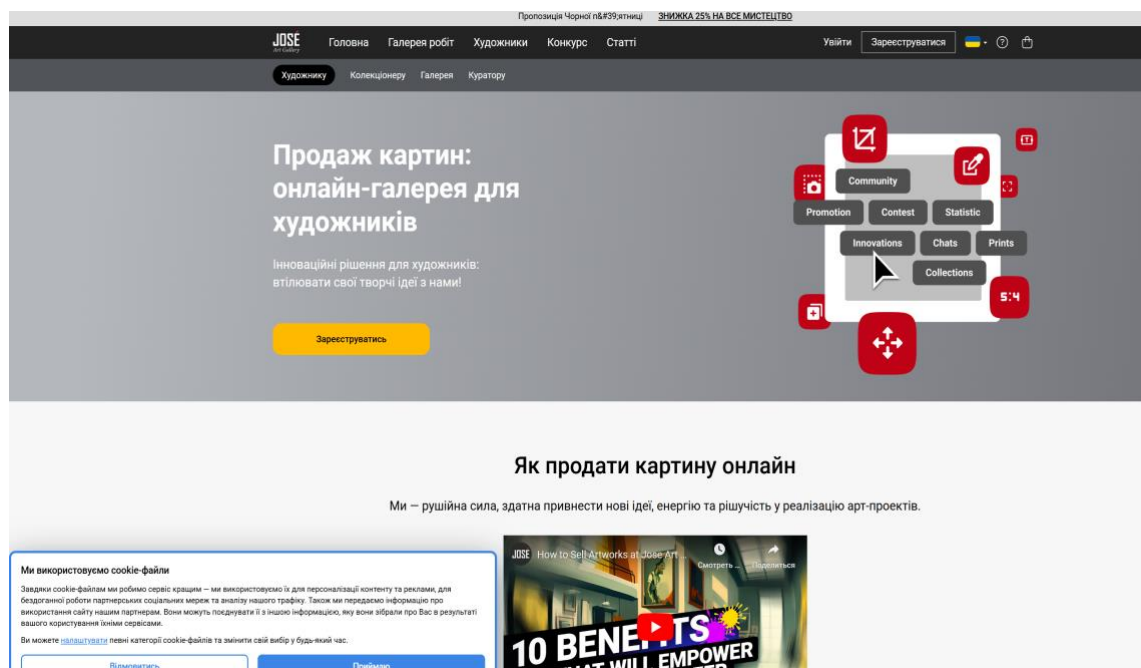


Рисунок 1.3 – Головна сторінка ресурсу Jose Art Gallery

На відміну від попередніх вебресурсів, Jose Art Gallery орієнтований не лише на покупців, а й на художників, які бажають продавати власні роботи через платформу. Саме тому на головній сторінці значна увага приділена реєстрації авторів та популяризації можливостей співпраці з сервісом.

Ресурс містить великий обсяг інформації, що охоплює продаж картин, конкурси, статті, галереї, колекції та інші сервіси. З одного боку, це розширює функціональні можливості системи, проте з іншого — ускладнює сприйняття інтерфейсу новими користувачами.

На головній сторінці присутня значна кількість інтерактивних елементів, кнопок та інформаційних блоків. Через це увага користувача може розсіюватися між різними елементами інтерфейсу.

					КВРІПЗ.220196.01.03.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис.	Дата		19

До переваг Jose Art Gallery належать:

- широкий функціонал;
- міжнародна орієнтація платформи;
- підтримка художників та колекціонерів;
- велика кількість мистецьких категорій;
- розвинені можливості просування авторів;
- багатомовність ресурсу.

Недоліками можна даного вебресурсу ми вважаємо:

- велика перевантаженість головної сторінки інформацією;
- складнішу навігацію порівняно з іншими ресурсами;
- велику кількість інтерактивних елементів;
- необхідність більш тривалого ознайомлення з функціоналом сайту.

Для узагальнення результатів аналізу доцільно виконати порівняння досліджуваних вебресурсів за основними характеристиками.

Таблиця 1.1 – Порівняльна характеристика існуючих аналогів.

Характеристика	ArtSale.ua	Art-Holst	Jose Art Gallery
Пошук по сайту	Так	Так	Так
Каталог картин	Так	Так	Так
Категорії товарів	Так	Так	Так
Кошик покупок	Так	Так	Так
Реєстрація користувачів	Так	Так	Так
Онлайн-консультації	Так	Ні	Частково
Мінімалістичний дизайн	Середній	Так	Ні
Багатомовність	Так	Ні	Так
Особистий кабінет	Так	Так	Так
Орієнтація на художників	Частково	Ні	Так

Проведений аналіз показав, що жоден із розглянутих ресурсів не є повністю універсальним рішенням. Кожен вебсайт має власні переваги та недоліки. ArtSale.ua характеризується зручним функціоналом та широкими можливостями взаємодії з користувачем. Art-Holst демонструє вдале поєднання простоти та естетичності інтерфейсу. Jose Art Gallery відзначається широкими функціональними можливостями та орієнтацією на міжнародний ринок.

На основі проведеного аналізу можна зробити висновок, що під час розроблення вебсайту онлайн-галереї з продажу картин доцільно поєднати переваги розглянутих аналогів. Майбутня наша система повинна мати зрозумілий інтерфейс, ефективний механізм пошуку, зручний каталог картин, можливість реєстрації користувачів, кошик покупок, систему оформлення замовлень та адаптивний дизайн. Це дозволить створити конкурентоспроможний програмний продукт, який забезпечуватиме комфортну взаємодію між покупцями та продавцями творів мистецтва.

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

Одним із найважливіших етапів розроблення будь-якого програмного забезпечення є визначення вимог до майбутньої системи. Саме вимоги визначають функціональні можливості програмного продукту, особливості його роботи, критерії якості та обмеження, яких необхідно дотримуватись під час проектування і реалізації.

Вебсайт онлайн-галереї з продажу картин є складною інформаційною системою, яка забезпечує взаємодію між продавцем та покупцем через мережу Інтернет. Основною метою створення такого вебресурсу є автоматизація процесів демонстрації художніх творів, пошуку необхідних товарів, оформлення замовлень та здійснення онлайн-покупок.

					КвРІПЗ.220196.01.03.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис.	Дата		21

Для формування вимог було виконано аналіз предметної області, досліджено існуючі аналоги та визначено потреби потенційних користувачів системи. Результатом цього аналізу стало формування функціональних та нефункціональних вимог до програмного забезпечення.

### 1.3.1 Функціональні вимоги

Функціональні вимоги визначають перелік функцій, які повинна виконувати система для забезпечення потреб користувачів та досягнення поставлених цілей.

Основними користувачами вебсайту є зареєстровані відвідувачі, нереєстровані користувачі та адміністратор системи.

Кожна з цих категорій користувачів повинна мати власний набір доступних функцій.

Незареєстрований користувач повинен мати можливість:

- переглядати головну сторінку сайту;
- переглядати каталог картин;
- здійснювати пошук картин за ключовими словами;
- використовувати систему фільтрації;
- переглядати детальну інформацію про картини;
- переглядати інформацію про художників;
- переглядати умови доставки та оплати;
- проходити процедуру реєстрації;
- переходити до сторінки авторизації.

Каталог картин повинен бути доступним для перегляду без необхідності реєстрації, оскільки це дозволяє потенційним покупцям ознайомитися з асортиментом продукції та оцінити можливості вебресурсу.

До функцій зареєстрованого користувача відноситься те, що після проходження процедури авторизації користувач отримує розширені можливості роботи із системою.

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						22
Змін.	Арк.	№ докум.	Підпис.	Дата		

Загалом зареєстрований користувач повинен мати такі можливості:

- виконувати вхід до особистого кабінету;
- редагувати персональні дані;
- змінювати пароль доступу;
- переглядати історію замовлень;
- додавати картини до кошика;
- видаляти товари з кошика;
- змінювати кількість товарів у кошику;
- оформлювати замовлення;
- обирати спосіб доставки;
- обирати спосіб оплати;
- переглядати статус виконання замовлення;
- отримувати повідомлення про успішне оформлення замовлення;
- додавати картини до списку обраних товарів.

Особистий кабінет користувача повинен містити інформацію про всі раніше оформлені замовлення, що дозволить покупцю відстежувати історію своїх покупок.

До функцій адміністратора відноситься те, що він є головним користувачем системи, який відповідає за наповнення та підтримку вебресурсу.

Адміністратор повинен мати такі можливості та повноваження:

- виконувати авторизацію в адміністративній панелі;
- додавати нові картини;
- редагувати інформацію про картини;
- видаляти картини з каталогу;
- створювати нові категорії;
- редагувати категорії;
- видаляти категорії;
- переглядати список користувачів;
- керувати обліковими записами користувачів;

					КвРІПЗ.220196.01.03.ІЗ	Арк.
						23
Змін.	Арк.	№ докум.	Підпис.	Дата		

- переглядати замовлення;
- змінювати статус замовлень;
- переглядати статистичну інформацію;
- керувати вмістом вебсайту;
- додавати новини та інформаційні матеріали.

Для забезпечення коректної роботи системи адміністратор повинен мати доступ до всіх основних функціональних модулів вебресурсу.

### 1.3.2 Вимоги щодо каталогу картин

Каталог картин є центральним компонентом системи. Для кожної картини повинна зберігатися така інформація:

- назва картини;
- ім'я митця;
- категорія;
- жанр;
- стиль виконання;
- рік створення картини;
- розміри картини;
- матеріали виконання;
- ціна картини;
- опис картинни;
- фотографії;
- статус наявності картинни.

Проектована система повинна забезпечувати можливість швидкого пошуку картин за різними критеріями. Так, система пошуку повинна підтримувати:

- пошук за назвою;
- пошук за автором;
- пошук за категорією;

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						24
Змін.	Арк.	№ докум.	Підпис.	Дата		

- пошук за ключовими словами.

Фільтрація каталогу повинна здійснюватися за такими параметрами:

- ціна картини;
- художник;
- стиль;
- жанр;
- популярність художника;
- дата додавання;
- наявність картини.

Результати пошуку повинні відображатися у вигляді впорядкованого списку із можливістю сортування користувачем.

### 1.3.3 Вимоги до оформлення замовлення

Система повинна забезпечувати повний цикл оформлення замовлення через вебінтерфейс. Процес оформлення замовлення повинен включати:

- Додавання картини до кошика.
- Перегляд складу замовлення.
- Введення контактної інформації.
- Вибір способу доставки.
- Вибір способу оплати.
- Підтвердження замовлення.
- Надсилання повідомлення користувачу.

Після успішного оформлення замовлення система повинна автоматично формувати запис у базі даних та змінювати статус товару за необхідності. Користувач повинен отримувати повідомлення про успішне оформлення замовлення на електронну пошту.

### 1.3.4 Вимоги до бази даних

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						25
Змін.	Арк.	№ докум.	Підпис.	Дата		

Для зберігання інформації необхідно використовувати реляційну базу даних.

База даних повинна забезпечувати зберігання інформації про:

- користувачів;
- картини;
- категорії;
- художників;
- замовлення;
- платежі;
- повідомлення;
- відгуки.

Усі дані повинні зберігатися централізовано та бути доступними відповідно до прав доступу користувачів.

База даних нашого застосунку повинна забезпечувати:

- цілісність інформації;
- захист від втрати даних;
- швидке виконання запитів;
- масштабованість системи.

### 1.3.5 Нефункціональні вимоги

Нефункціональні вимоги визначають характеристики якості програмного забезпечення та умови його функціонування, зокрема вимоги до продуктивності, вимоги до надійності та безпеки, вимоги до зручності використання, вимоги до адаптивності та вимоги до сумісності

Так, система повинна забезпечувати високу швидкість роботи незалежно від кількості користувачів. Час завантаження основних сторінок вебсайту не повинен перевищувати 3 секунд за стандартних умов роботи. Час виконання пошукового запиту не повинен перевищувати 2 секунд.

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						26
Змін.	Арк.	№ докум.	Підпис.	Дата		

Також, система повинна підтримувати одночасну роботу великої кількості користувачів без суттєвого зниження продуктивності.

Відповідно до вимог надійності програмне забезпечення повинно забезпечувати стабільну роботу протягом тривалого часу. Зокрема, система повинна:

- коректно обробляти помилки;
- запобігати втраті даних;
- підтримувати резервне копіювання;
- забезпечувати відновлення після збоїв.

У разі виникнення помилки користувач повинен отримувати зрозуміле повідомлення без відображення службової інформації системи.

Одним із найважливіших аспектів функціонування вебсайту є забезпечення захисту персональних даних користувачів. Тут система повинна забезпечувати:

- захищену авторизацію користувачів;
- шифрування паролів;
- використання протоколу HTTPS;
- захист від SQL-ін'єкцій;
- захист від міжсайтового скриптингу (XSS);
- захист від CSRF-атак;
- розмежування прав доступу користувачів.

Особисті дані користувачів повинні використовуватися виключно для забезпечення роботи вебресурсу та не можуть передаватися третім особам без відповідної згоди.

Інтерфейс вебсайту повинен бути інтуїтивно зрозумілим та не вимагати спеціальної підготовки користувача. При розробленні інтерфейсу необхідно дотримуватись таких принципів:

- простота навігації;
- логічне розташування елементів;
- зрозумілі назви пунктів меню;

					КвРІПЗ.220196.01.03.ІЗ	Арк.
Змін.	Арк.	№ докум.	Підпис.	Дата		27

- мінімальна кількість дій для виконання операцій;
- єдиний стиль оформлення сторінок.

Важливим критерієм є дотримання принципу «трьох кліків», відповідно до якого користувач повинен мати можливість знайти необхідну інформацію не більше ніж за три переходи між сторінками.

Вебсайт повинен коректно працювати на різних типах пристроїв, таких як персональний комп'ютер, ноутбук, планшет, смартфон. Таким чином, інтерфейс повинен автоматично адаптуватися до різних розмірів екрана без втрати функціональності.

Програмний продукт повинен підтримувати роботу у найбільш поширених веббраузерах Google Chrome, Mozilla Firefox, Microsoft Edge, Opera, Safari.

Також коректне відображення сторінок повинно забезпечуватися незалежно від операційної системи користувача.

#### 1.4 Висновки. Постановка задачі

У результаті аналізу предметної області були сформовані функціональні та нефункціональні вимоги до вебсайту онлайн-галереї з продажу картин. Визначено основні категорії користувачів системи, перелік їх можливостей та вимоги до основних функціональних модулів. Окрему увагу приділено питанням безпеки, продуктивності, адаптивності та зручності використання. Сформовані вимоги будуть використані на наступних етапах проектування програмного забезпечення для розроблення архітектури системи та реалізації її функціональних можливостей.

Основними задачами для виконання роботи є:

- проведення аналізу сучасних технологій розробки вебзастосунків та електронної комерції;
- визначення функціональних вимог до системи онлайн-продажу картин;
- проектування структури бази даних і архітектури програмного забезпечення;
- розробка адаптивного вебінтерфейсу користувача;

					КвРІПЗ.220196.01.03.ІЗ	Арк.
						28
Змін.	Арк.	№ докум.	Підпис.	Дата		

- реалізація функціоналу онлайн-галереї з каталогом картин та системою замовлень;
- створення адміністративної панелі керування контентом;
- тестування та перевірка працездатності розробленого програмного продукту.

					КВРІПЗ.220196.01.03.ПЗ	Арк.
						29
Змін.	Арк.	№ докум.	Підпис.	Дата		

## 2. ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1 Вибір типу архітектури та шаблонів проєктування

Архітектура програмного забезпечення є одним із найважливіших етапів розроблення інформаційної системи. Вона визначає загальну структуру програмного продукту, принципи взаємодії його компонентів, механізми обробки даних та особливості функціонування системи в цілому. Вірно та грамотно спроектована архітектура забезпечує високу продуктивність, масштабованість, надійність та простоту подальшого супроводу програмного забезпечення.

Під час розроблення вебсайту онлайн-галереї з продажу картин особлива увага приділялася побудові архітектури, яка забезпечувала б ефективну взаємодію між користувачами, серверною частиною та базою даних. Враховуючи функціональні вимоги, сформовані в попередньому розділі, було прийнято рішення використовувати клієнт-серверну архітектуру, яка є найбільш поширеним підходом до створення сучасних веборієнтованих інформаційних систем.

Клієнт-серверна архітектура передбачає розподіл системи на дві основні складові: клієнтську та серверну частини. Клієнтська частина відповідає за взаємодію з користувачем, відображення інформації та формування запитів до сервера. Серверна частина забезпечує обробку отриманих запитів, виконання бізнес-логіки, взаємодію з базою даних та формування відповідей клієнту.

Основними перевагами клієнт-серверної архітектури є:

- централізоване зберігання даних;
- можливість одночасної роботи великої кількості користувачів;
- підвищений рівень безпеки;
- спрощення адміністрування системи;
- зручність оновлення програмного забезпечення;
- можливість подальшого масштабування проєкту.

Для розроблюваного вебсайту така архітектура є оптимальною, оскільки всі дані про картини, користувачів, замовлення та платежі повинні зберігатися

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						30
Змін.	Арк.	№ докум.	Підпис.	Дата		

централізовано та бути доступними кожному незалежно від місця знаходження конкретного користувача.

Загальний принцип функціонування системи полягає у тому, що користувач за допомогою веббраузера відкриває вебсайт онлайн-галереї та здійснює взаємодію з його інтерфейсом. Під час виконання певних дій, таких як пошук картини, авторизація, оформлення замовлення або перегляд особистого кабінету, браузер надсилає відповідний HTTP-запит на сервер. Сервер обробляє отриманий запит, звертається до бази даних у разі необхідності та повертає сформовану відповідь клієнту.

Архітектура програмного забезпечення включає такі основні компоненти, як клієнтський рівень, рівень бізнес-логіки, рівень доступу до даних та систему керування базою даних.

Клієнтський рівень представлений вебінтерфейсом користувача. Саме цей компонент відповідає за відображення інформації на екрані, взаємодію з меню навігації, каталогом картин, формами авторизації та оформлення замовлення.

Рівень бізнес-логіки забезпечує реалізацію основних функцій системи. До них належать:

- керування користувачами;
- пошук та фільтрація картин;
- обробка замовлень;
- керування кошиком покупок;
- перевірка коректності введених даних;
- взаємодія з платіжними сервісами.

Рівень доступу до даних відповідає за обмін інформацією між серверною частиною та базою даних. Він забезпечує виконання запитів на отримання, створення, оновлення та видалення інформації.

База даних виступає центральним сховищем інформації системи. У ній зберігаються відомості про користувачів, картини, художників, категорії, замовлення та інші об'єкти предметної області.

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						31
Змін.	Арк.	№ докум.	Підпис.	Дата		

Важливим елементом архітектури є механізм автентифікації та авторизації користувачів. Для забезпечення безпеки доступу до системи користувач повинен пройти процедуру входу за допомогою логіна та пароля. Після успішної перевірки облікових даних система надає відповідний рівень доступу до функціональних можливостей вебресурсу.

У межах проекту передбачено використання трьох основних ролей користувачів, таких неавторизований користувач, авторизований користувач та адміністратор.

Авторизовані користувачі мають можливість переглядати каталог картин, здійснювати пошук, додавати товари до кошика та оформлювати замовлення.

Адміністратор отримує додаткові можливості, пов'язані з керуванням контентом вебсайту. Він може додавати нові картини, редагувати існуючі записи, переглядати замовлення та керувати категоріями товарів.

Для моделювання функціональних можливостей системи було створено діаграму варіантів використання (рисунок 2.1).

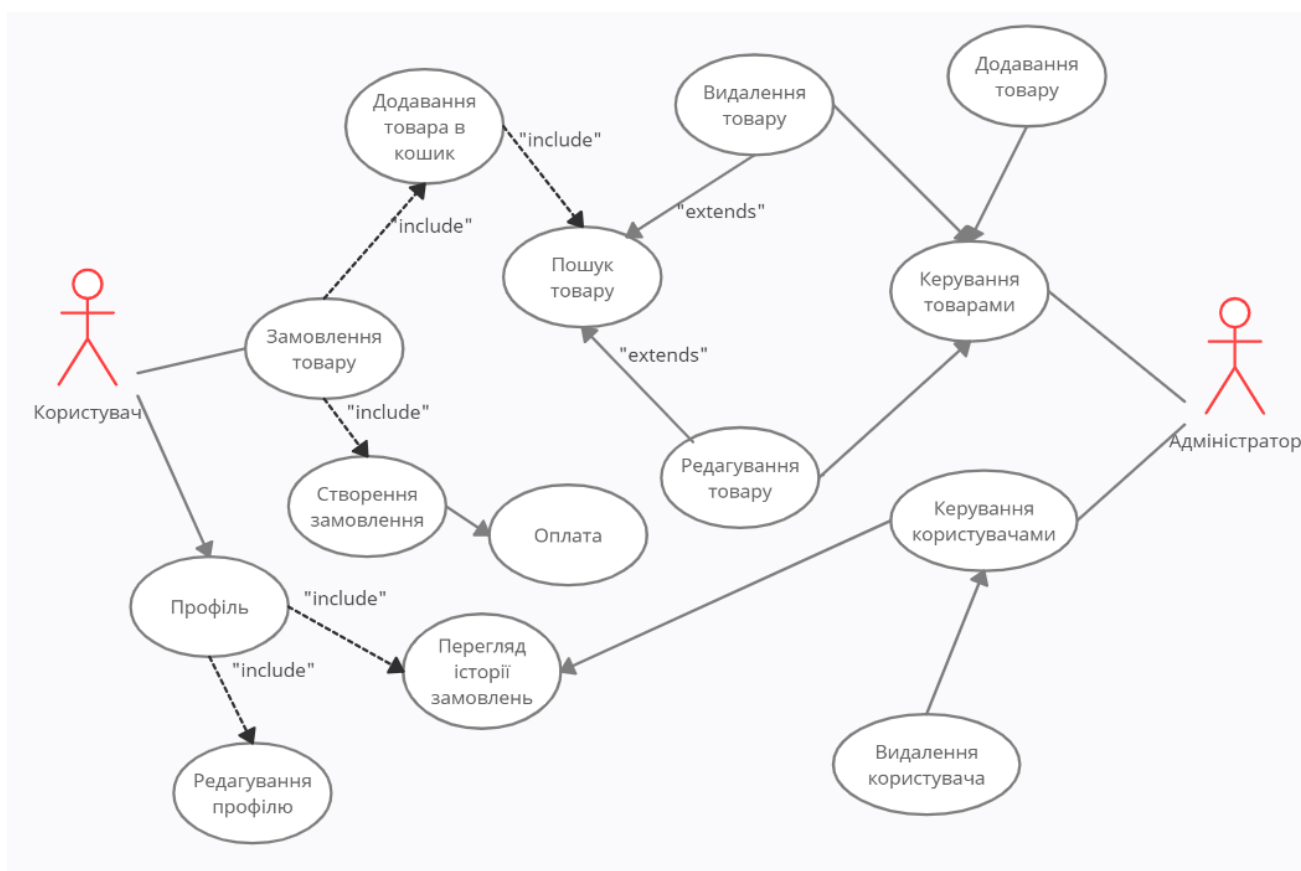


Рисунок 2.1 – Діаграма Use Case

На діаграмі представлені основні учасники системи та сценарії їх взаємодії з вебресурсом. Основними акторами виступають користувач та адміністратор.

Авторизований користувач взаємодіє із системою через такі варіанти використання:

- реєстрація;
- авторизація;
- перегляд каталогу;
- пошук картин;
- перегляд інформації про картину;
- додавання товару до кошика;
- оформлення замовлення;
- редагування профілю.

Адміністратор взаємодіє із системою через наступні функції:

- авторизація;
- додавання картин;
- редагування картин;
- видалення картин;
- керування категоріями;
- перегляд замовлень;
- зміна статусів замовлень.

Наступним етапом моделювання стала побудова діаграми станів користувача (рисунок 2.2).

Діаграма станів відображає можливі стани користувача під час роботи із системою та переходи між ними. Основними станами є відвідування сайту, авторизація, перегляд каталогу, додавання товарів до кошика, оформлення замовлення та завершення сеансу роботи.

Використання UML-діаграм дозволяє формалізувати логіку роботи системи ще до початку програмної реалізації та зменшити ймовірність помилок під час розроблення програмного забезпечення.

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						33
Змін.	Арк.	№ докум.	Підпис.	Дата		

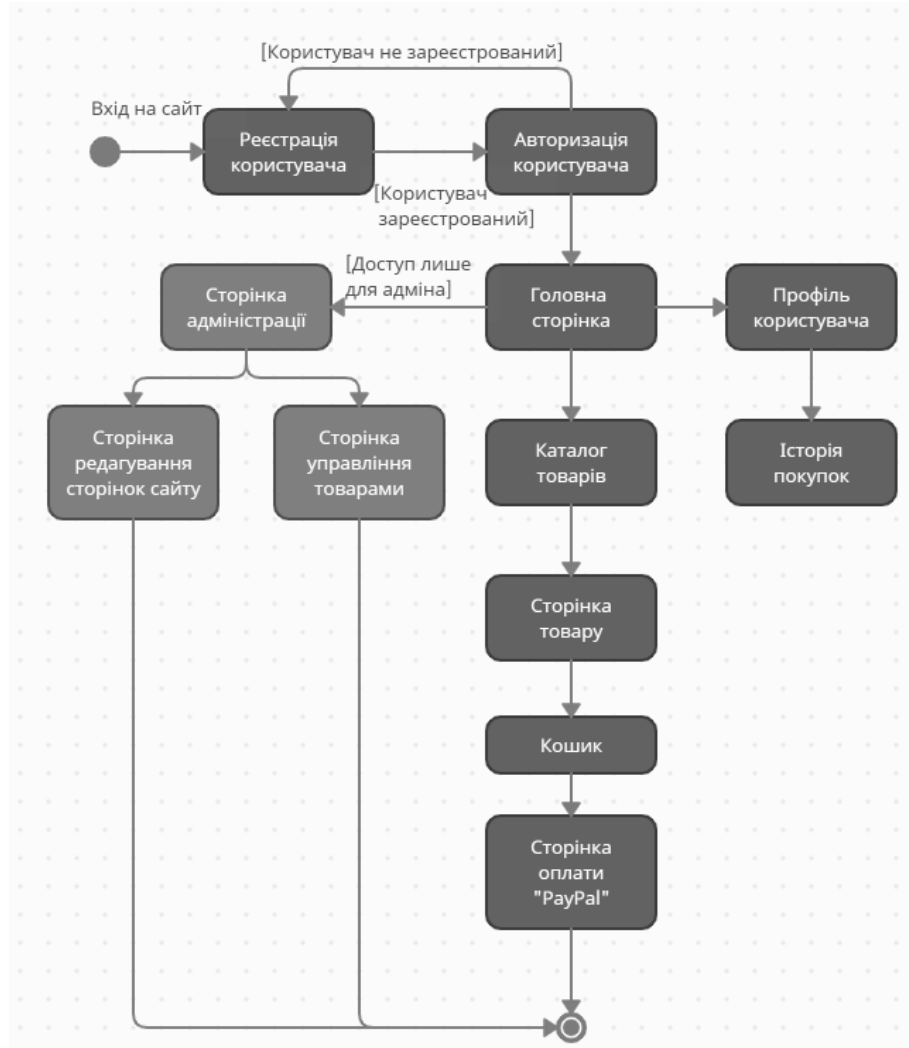


Рисунок 2.2 – Діаграма станів

Таким чином, розроблена архітектура забезпечує логічний розподіл функцій між компонентами системи, централізоване зберігання інформації, безпечну взаємодію користувачів із вебресурсом та можливість подальшого розвитку програмного продукту.

## 2.2 Детальне проєктування програмного забезпечення

Після визначення загальної архітектури системи переходимо до наступного етапу розроблення, а саме детального проєктування програмного застосунку. На даному етапі виконаємо моделювання взаємодії компонентів системи, аналіз

					КвРІПЗ.220196.01.03.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис.	Дата		34

потоків даних, побудова UML-діаграм та визначення механізмів реалізації основних бізнес-процесів.

Детальне проєктування дозволяє нам формалізувати логіку роботи програмного забезпечення ще до початку програмування, що суттєво знижує ризик виникнення помилок під час реалізації проєкту. Крім того, наявність детально опрацьованої моделі системи значно спрощує її подальше тестування, супровід та модернізацію.

Для вебсайту онлайн-галереї з продажу картин було виконано моделювання основних процесів взаємодії користувачів із системою, а також розроблено діаграми, які відображають механізми обміну інформацією між окремими компонентами програмного комплексу.

### 2.2.1 Проєктування клієнт-серверної взаємодії

В основі роботи вебсайту лежить клієнт-серверна модель взаємодії. Такий підхід передбачає розподіл функцій між клієнтською частиною, яка працює безпосередньо у браузері користувача, та серверною частиною, яка виконує обробку даних та забезпечує взаємодію з базою даних.

Загальна схема клієнт-серверної взаємодії наведена на рисунку 2.3.

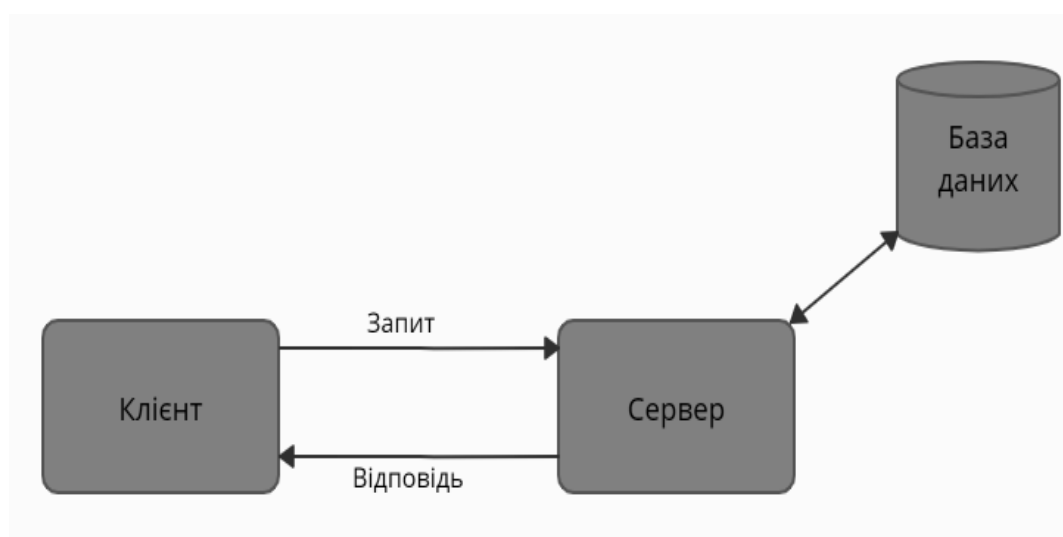


Рисунок 2.3 – Клієнт-серверна архітектура.

Користувач взаємодіє із системою за допомогою веббраузера. Під час виконання будь-якої дії формується HTTP-запит, який надсилається на сервер. Сервер виконує необхідну бізнес-логіку, звертається до бази даних та повертає результат обробки у вигляді вебсторінки або набору даних.

Послідовність обробки запиту включає такі етапи:

1. Формування запиту користувачем.
2. Надсилання запиту на сервер.
3. Обробка запиту контролером.
4. Виконання бізнес-логіки.
5. Отримання або зміна інформації у базі даних.
6. Формування відповіді.
7. Передача результату клієнту.
8. Відображення інформації користувачеві.

Такий підхід дозволяє забезпечити централізоване управління даними та підвищити рівень безпеки системи.

## 2.2.2 Проектування сценаріїв роботи користувача

Одним із найважливіших етапів моделювання є опис послідовності взаємодії користувача із системою.

Для цього нами було побудовано діаграму послідовності дій користувача, представлену на рисунку 2.4.

Діаграма послідовності використовується для відображення процесу взаємодії між користувачем та основними компонентами вебсайту онлайн-галереї з продажу картин. Вона дозволяє наочно продемонструвати порядок виконання операцій та обмін повідомленнями між учасниками системи в процесі виконання певного сценарію. На діаграмі відображено основні об'єкти системи, зокрема користувач, вебінтерфейс, контролер, серверна логіка та база даних. Процес починається із надсилання користувачем запиту через веббраузер. Після цього контролер обробляє запит та звертається до відповідної моделі.

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						36
Змін.	Арк.	№ докум.	Підпис.	Дата		

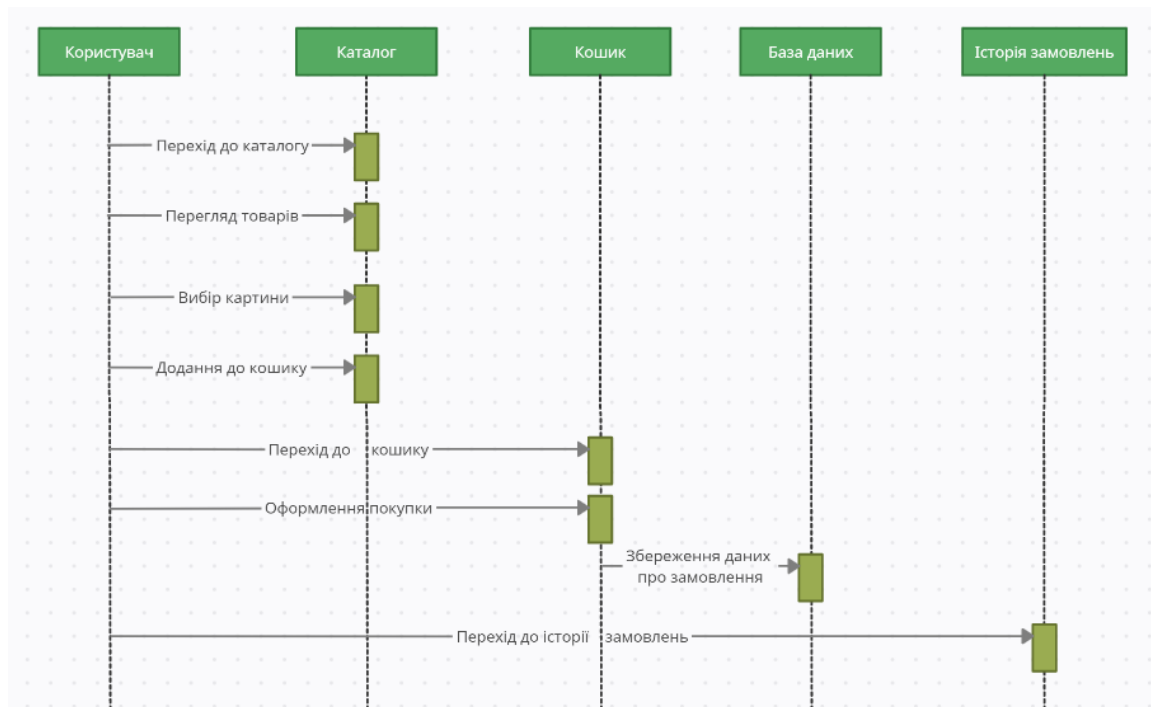


Рисунок 2.4 – Діаграма послідовності дій користувача

Дана діаграма відображає типовий сценарій роботи покупця з вебсайтом онлайн-галереї.

Після відкриття головної сторінки користувач отримує доступ до каталогу картин та навігаційного меню. За необхідності він може пройти процедуру реєстрації або авторизації.

Пошук необхідної картини може здійснюватися кількома способами:

- через каталог категорій;
- за допомогою пошукового рядка;
- через систему фільтрів;
- шляхом перегляду рекомендованих товарів.

Після вибору картини користувач переходить до сторінки товару, де може ознайомитися з детальними характеристиками твору мистецтва:

- назвою;
- автором;
- описом;
- розмірами;

- матеріалами виконання;
- вартістю;
- фотографіями.

Далі користувач додає обрану картину до кошика та переходить до оформлення замовлення.

На завершальному етапі здійснюється вибір способу доставки та способу оплати, після чого замовлення підтверджується та зберігається в базі даних

Інформація про оформлене замовлення автоматично надходить адміністратору для подальшої обробки.

### 2.2.3 Проєктування сценаріїв роботи адміністратора

Для забезпечення ефективного керування вебресурсом було передбачено окрему адміністративну панель.

Послідовність роботи адміністратора представлена на рисунку 2.5.

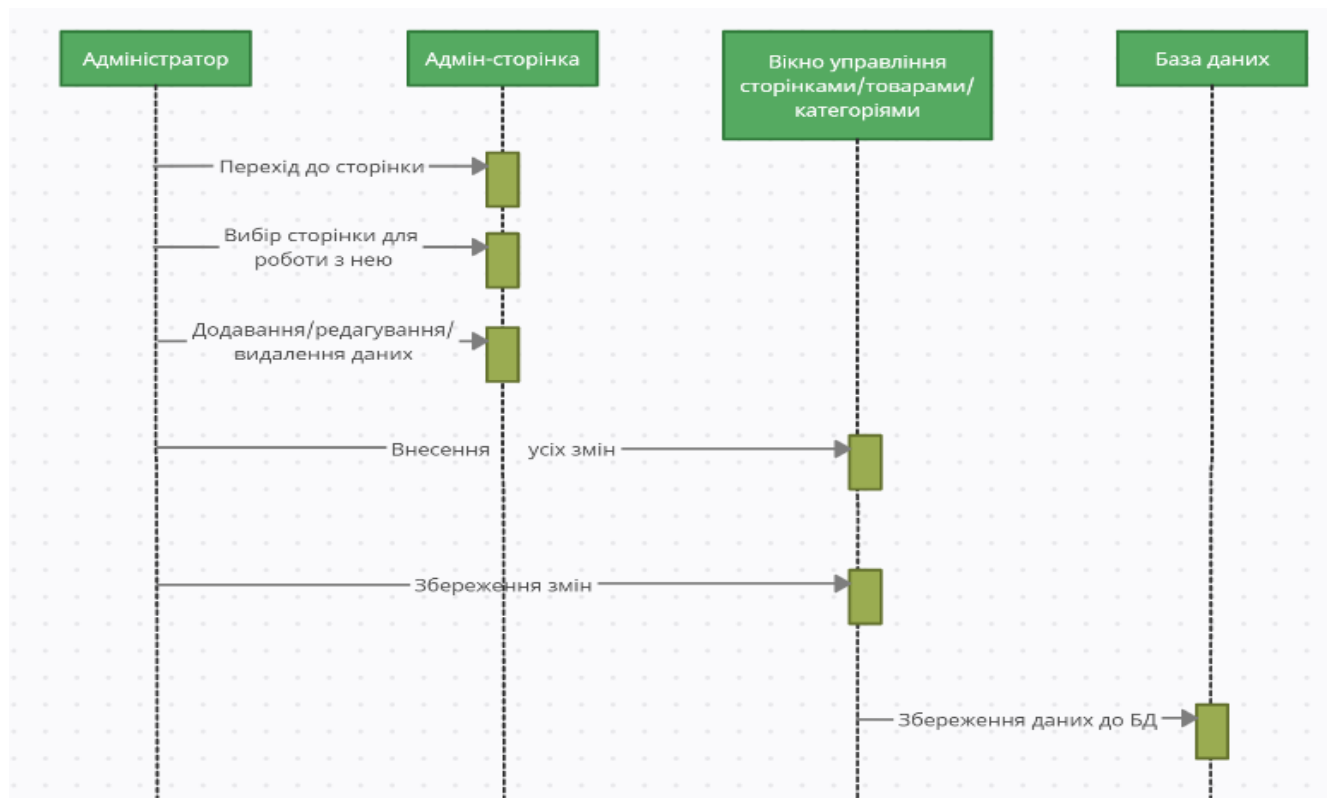


Рисунок 2.5 – Діаграма послідовності дій адміністратора

Перед початком роботи адміністратор проходить процедуру авторизації, використовуючи власний логін та пароль.

Після успішного входу в систему відкривається адміністративна панель, яка надає доступ до основних інструментів керування вебресурсом.

За допомогою адміністративної панелі можна виконувати такі операції:

- додавання нових картин;
- редагування існуючих записів;
- видалення картин;
- створення нових категорій;
- редагування категорій;
- перегляд замовлень;
- зміна статусів замовлень;
- перегляд інформації про користувачів.

Після завершення роботи адміністратор завершує сеанс та виходить із системи.

Такий підхід забезпечує високий рівень безпеки та дозволяє обмежити доступ до службових функцій стороннім користувачам.

#### 2.2.4 Проєктування механізму авторизації та реєстрації

Безпечна ідентифікація користувачів є важливою складовою будь-якої інформаційної системи.

Тому, для реалізації механізму реєстрації та авторизації нами було побудовано відповідну схему, наведену на рисунку 2.6.

Процес створення нового облікового запису складається з декількох етапів.

На першому етапі користувач вводить свої персональні дані:

- ім'я;
- електронну адресу;
- пароль;

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						39
Змін.	Арк.	№ докум.	Підпис.	Дата		



Схема даного бізнес-процесу представлена на рисунку 2.7.

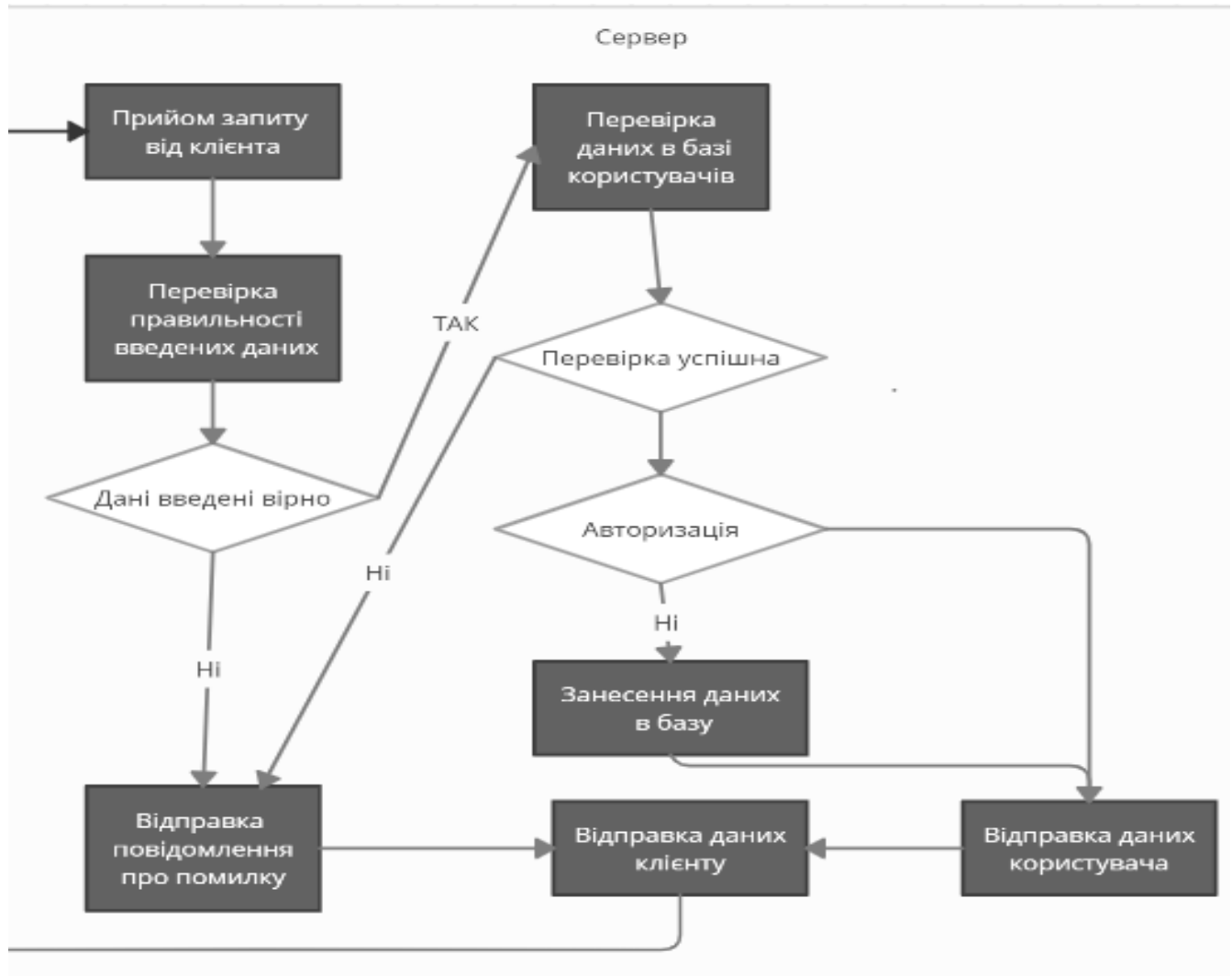


Рисунок 2.7 – Схема редагування профілю

Після авторизації користувач отримує доступ до особистого кабінету, де може змінювати власні персональні дані.

Процес редагування виконується у такій послідовності:

1. Користувач відкриває сторінку профілю.
2. Вносить необхідні зміни.
3. Натискає кнопку збереження.
4. Дані надсилаються на сервер.
5. Сервер виконує перевірку введеної інформації.
6. У разі успішної перевірки виконується оновлення запису у базі даних.
7. Користувач отримує повідомлення про успішне збереження змін.

Додатково система виконує перевірку прав доступу, що виключає можливість редагування профілів інших користувачів цього вебсайту для продажу картин.

Усі зміни зберігаються централізовано в базі даних та стають доступними під час наступного входу до системи.

### 2.3 Опис залежностей

Опис залежностей є етапом проектування, який дозволяє формалізувати взаємодію між компонентами системи, визначити ієрархію модулів та встановити шлях проходження потоків даних. У контексті розробки мобільного застосунку для обліку медикаментів правильне управління залежностями гарантує стабільність роботи пристрою в умовах обмежених ресурсів, запобігає виникненню критичних помилок під час роботи з локальною пам'яттю та забезпечує високу тестованість коду.

Архітектура застосунку базується на патерні MVVM, інтегрованому з принципами «чистої архітектури», що диктує суворе правило спрямування залежностей: від зовнішніх рівнів інтерфейсу та баз даних до внутрішніх рівнів абстракції та бізнес-моделей. Це означає, що компоненти, відповідальні за збереження фото чи планування сповіщень, можуть залежати від логіки опису препарату, проте ядро системи ніколи не містить знань про технічні особливості фреймворків SwiftData чи AVFoundation.

Шар користувацького інтерфейсу, реалізований за допомогою SwiftUI, функціонує як пасивний механізм відображення станів. Екрани списку ліків та форми додавання мають жорстку залежність виключно від відповідних класів ViewModel, які виступають джерелом істини для візуальних елементів. Графічний інтерфейс лише транслює дії користувача, такі як введення текстової дати або натискання кнопки камери, до рівня ViewModel, і підписується на оновлення станів для автоматичного рендерингу індикаторів придатності.

					КвРІПЗ.220196.01.03.ІЗ	Арк.
						42
Змін.	Арк.	№ докум.	Підпис.	Дата		

Класи ViewModel, у свою чергу, виступають інтелектуальним мостом, що зв'язує інтерфейс із рівнем даних. Вони абстраговані від безпосередньої реалізації сховищ і залежать лише від протоколів, що описують методи збереження та отримання інформації. Наприклад, логіка додавання нового препарату має залежність від абстрактного контракту сервісу сповіщень та репозиторію даних. Завдяки такій структурі досягається низька зв'язність коду, що дозволяє замінити механізм локального збереження на хмарну синхронізацію без необхідності вносити жодної зміни у код управління станом екрана.

Ядром системи виступає шар предметної області, представлений моделями SwiftData. Цей рівень є максимально незалежним і містить у собі опис сутності медикаменту, включаючи логіку розрахунку статусів придатності. Саме ці моделі диктують правила того, як саме атрибути «Назва», «Дата» та «Фото» повинні бути структуровані для подальшої обробки. Рівень даних реалізує ці моделі через механізм персистентності, де локальні репозиторії мають пряму залежність від контейнера бази даних для фіксації змін у SQLite.

Окремої уваги заслуговує інтеграція сервісних модулів камери та сповіщень. Модуль роботи з медіа залежить від системного API iOS, проте його взаємодія з основним кодом ініціюється виключно через інтерфейс провайдера даних. Перед тим як фотографія упаковки потрапляє до сховища, ViewModel звертається до сервісу обробки для оптимізації розміру файлу. Аналогічно, модуль сповіщень залежить від фреймворку UserNotifications, але отримує необхідні часові мітки від бізнес-логіки вже після успішної валідації текстової дати.

Для автоматизації управління життєвим циклом об'єктів у системі застосовано принцип інверсії управління. На відміну від складних зовнішніх фреймворків, у нативному середовищі Swift це реалізовано через механізм впровадження залежностей у конструктори та використання контейнерів середовища (Environment). Це дозволяє системі динамічно надавати екземпляр бази даних або сервісу камери тим компонентам, які їх потребують, без створення жорстких зв'язків між класами.

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						43
Змін.	Арк.	№ докум.	Підпис.	Дата		

Завдяки такому підходу система поділена на ізольовані напрямки постачання залежностей. Системний модуль забезпечує глобальний доступ до контексту SwiftData, гарантуючи єдину точку входу для всіх операцій з базою даних. Сервісний модуль керує ініціалізацією камери та налаштуванням дозволів операційної системи. Модуль сповіщень забезпечує зв'язок між бізнес-правилами придатності ліків та системною чергою нагадувань. Зрештою, навігаційний модуль координує переходи між екранами, передаючи необхідні залежності від батьківських компонентів до дочірніх, що забезпечує цілісність архітектурної побудови застосунку.

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

Одним із найважливіших етапів розроблення програмного забезпечення є вибір технологій та програмних засобів, які будуть використовуватися під час реалізації проєкту. Від правильності цього вибору залежить продуктивність системи, швидкість розробки, можливість подальшого супроводу та масштабування програмного продукту.

Під час створення вебсайту онлайн-галереї з продажу картин нами було проведено аналіз сучасних технологій веброботки, засобів програмування, систем управління базами даних та програмних платформ. За результатами проведеного дослідження було обрано технологічний стек, який найбільш повно відповідає поставленим вимогам та забезпечує реалізацію всіх необхідних функціональних можливостей системи.

Основними критеріями вибору технологій були надійність, продуктивність, масштабованість, безпечність, зручність розробки, підтримка сучасних вебстандартів, можливість інтеграції з базами даних, підтримка архітектури MVC.

Для реалізації проєкту було обрано платформу Microsoft .NET та технологію ASP.NET Core MVC.

					КВРІПЗ.220196.01.03.ПЗ	Арк.
						44
Змін.	Арк.	№ докум.	Підпис.	Дата		

### 2.3.1 Вибір середовища розробки

Для розроблення програмного забезпечення використовується інтегроване середовище розробки Visual Studio Code

Visual Studio Code є одним із найпопулярніших середовищ розробки програмного забезпечення у світі. Воно забезпечує повний набір інструментів для створення, тестування, налагодження та супроводу програмних продуктів різного рівня складності.

Серед основних переваг Visual Studio Code можна виділити:

- підтримку великої кількості мов програмування;
- автоматичне доповнення коду IntelliSense;
- вбудовані інструменти налагодження;
- інтеграцію із системами контролю версій;
- підтримку роботи з базами даних;
- можливість автоматичного тестування;
- підтримку сучасних вебтехнологій.

Використання Visual Studio Code дозволяє значно прискорити процес розроблення вебзастосунку та підвищити якість програмного коду та його повторне використання.

Середовище також містить інструменти для аналізу продуктивності додатків, профілювання коду та виявлення потенційних помилок ще на етапі програмування.

Завдяки широкому поширенню Visual Studio існує велика кількість документації, навчальних матеріалів та прикладів реалізації різноманітних програмних рішень.

### 2.3.2 Вибір платформи ASP.NET Core

Для створення серверної частини вебсайту нами було обрано платформу ASP.NET Core.

					КВРІПЗ.220196.01.03.ПЗ	Арк.
						45
Змін.	Арк.	№ докум.	Підпис.	Дата		

ASP.NET Core є сучасним фреймворком для розроблення вебзастосунків, створеним компанією Microsoft. Він входить до складу платформи .NET та дозволяє створювати високопродуктивні вебсистеми різного рівня складності.

Основними перевагами ASP.NET Core є:

- висока швидкодія;
- кросплатформеність;
- модульна архітектура;
- вбудовані механізми безпеки;
- підтримка шаблону MVC;
- інтеграція з Entity Framework Core;
- підтримка REST API;
- можливість розгортання у хмарних середовищах.

Використання ASP.NET Core дозволяє створювати сучасні вебсайти, які можуть працювати на операційних системах Windows, Linux та macOS.

Особливістю платформи є підтримка механізму Dependency Injection, який забезпечує слабе зв'язування між компонентами системи та спрощує подальше тестування програмного забезпечення.

Ще однією важливою перевагою ASP.NET Core є можливість реалізації багаторівневої архітектури програмного забезпечення, що відповідає вимогам до сучасних інформаційних систем.

### 2.3.3 Використання архітектурного шаблону MVC

Для побудови структури вебзастосунку було використано архітектурний шаблон Model-View-Controller (MVC).

MVC є одним із найпоширеніших архітектурних патернів при розробленні вебдодатків. Його головною метою є розділення функціональності програмного забезпечення на окремі компоненти.

					КвРІПЗ.220196.01.03.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис.	Дата		46

Структура MVC складається з трьох основних елементів Model, View та Controller. Загальна схема роботи патерну MVC наведена на рисунку 2.8.

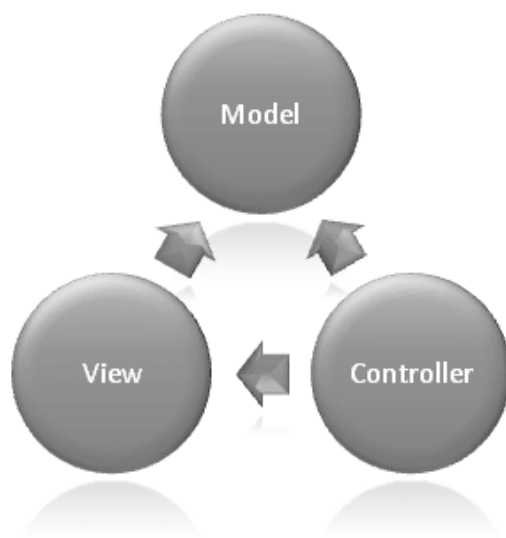


Рисунок 2.8 – Схема взаємодії компонентів MVC

Модель відповідає за зберігання та обробку даних предметної області. Саме в моделі реалізується бізнес-логіка додатку.

Загалом для вебсайту онлайн-галереї до моделей відносяться користувачі, картини, категорії, замовлення, платежі, відгуки.

Представлення відповідає за відображення інформації користувачеві. У нашому проєкті представлення реалізуються за допомогою технології Razor Views та мови розмітки HTML.

Контролер виступає посередником між користувачем та системою. Основними функціями контролерів є прийом запитів, перевірка даних, виклик бізнес-логіки, формування відповіді, передача даних у представлення.

Основними перевагами використання MVC є:

- розділення відповідальностей між компонентами;
- кожен компонент відповідає за свою роботу і пов'язаний з іншими;
- простота супроводу системи;
- покращення масштабованості;
- зручність тестування;

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						47
Змін.	Арк.	№ докум.	Підпис.	Дата		

- повторне використання коду.

Саме тому даний архітектурний підхід був обраний для реалізації вебсайту онлайн-галереї.

#### 2.3.4 Вибір системи керування базами даних

Для зберігання інформації про користувачів, картини та замовлення необхідно використовувати систему керування базами даних. У межах даного проєкту нами було обрано Microsoft SQL Server. Microsoft SQL Server є реляційною системою управління базами даних, яка забезпечує:

- надійне зберігання інформації;
- високу швидкість виконання запитів;
- підтримку великих обсягів даних;
- механізми резервного копіювання;
- засоби забезпечення безпеки.

До основних переваг SQL Server належать:

- висока продуктивність;
- інтеграція з платформою .NET;
- підтримка транзакцій;
- забезпечення цілісності даних;
- підтримка механізмів реплікації.

У базі даних будуть зберігатися наступні сутності:

- Users;
- UserRoles;
- Roles;
- Paintings;
- Categories;
- Orders;
- OrderDetails;

					КвРІПЗ.220196.01.03.ІЗ	Арк.
						48
Змін.	Арк.	№ докум.	Підпис.	Дата		

– Pages.

Використання реляційної моделі даних дозволяє забезпечити ефективний зв'язок між усіма сутностями системи.

### 2.3.5 Використання Entity Framework Core

Для організації взаємодії між програмним кодом та базою даних було обрано технологію Entity Framework Core.

Entity Framework Core є ORM-фреймворком (Object Relational Mapping), який дозволяє працювати з базою даних через об'єкти мови програмування високого рівня C#.

Використання ORM дозволяє уникнути написання великої кількості SQL-запитів вручну.

Основними перевагами Entity Framework Core є:

- автоматичне створення таблиць бази даних;
- підтримка міграцій;
- швидке виконання CRUD-операцій;
- інтеграція з ASP.NET Core;
- зменшення кількості програмного коду.

Використання даної технології значно прискорює розроблення програмного забезпечення та спрощує подальшу підтримку системи.

### 2.3.6 Технології клієнтської частини

Для реалізації інтерфейсу користувача використовуються сучасні вебтехнології.

Основою побудови сторінок виступає HTML5. HTML5 дозволяє створювати структуровані вебдокументи та забезпечує правильне відображення інформації у браузері.

					КВРІПЗ.220196.01.03.ПЗ	Арк.
						49
Змін.	Арк.	№ докум.	Підпис.	Дата		

Для оформлення зовнішнього вигляду використовується CSS3. За допомогою CSS реалізуються, кольорове оформлення, розташування елементів, адаптивний дизайн, анімації, візуальні ефекти тощо.

Для забезпечення інтерактивності використовується JavaScript. JavaScript, в свою чергу дозволяє:

- перевіряти введені дані;
- виконувати асинхронні запити;
- оновлювати сторінки без перезавантаження;
- створювати динамічні елементи інтерфейсу.

Для пришвидшення розробки адаптивного інтерфейсу використовується фреймворк Bootstrap. Він забезпечує:

- адаптацію сторінок під мобільні пристрої;
- готові елементи інтерфейсу;
- систему сіток;
- стандартизований зовнішній вигляд сторінок.

Завдяки використанню Bootstrap вебсайт коректно відображається на комп'ютерах, планшетах та смартфонах.

### 2.3.7 Засоби забезпечення безпеки

Оскільки система працює з персональними даними користувачів, питання безпеки є надзвичайно важливим.

Для захисту інформації в нашомук проекті використовуються:

- HTTPS-протокол;
- шифрування паролів;
- механізми авторизації ASP.NET Identity;
- перевірка введених даних;
- захист від SQL-ін'єкцій;
- захист від XSS-атак;

					КвРІПЗ.220196.01.03.ІЗ	Арк.
						50
Змін.	Арк.	№ докум.	Підпис.	Дата		

– захист від CSRF-запитів.

Застосування сучасних механізмів захисту дозволяє забезпечити безпечну роботу вебресурсу та збереження персональних даних користувачів.

## 2.4 Висновки другого розділу

У другому розділі дипломної роботи було виконано проектування програмного забезпечення вебсайту онлайн-галереї з продажу картин та обґрунтовано вибір методів і засобів його реалізації.

У процесі проектування було визначено загальну архітектуру програмної системи. Для реалізації вебресурсу обрано клієнт-серверну архітектуру, яка забезпечує централізоване зберігання даних, ефективну взаємодію між користувачами та сервером, високий рівень безпеки й можливість масштабування програмного продукту. Даний підхід дозволяє розподілити функціональність між клієнтською та серверною частинами системи, що позитивно впливає на продуктивність та зручність подальшого супроводу програмного забезпечення.

Під час детального проектування було змодельовано основні бізнес-процеси вебсайту та побудовано UML-діаграми, які відображають логіку функціонування системи. Зокрема, було розроблено діаграму варіантів використання, діаграму станів користувача, діаграми послідовності дій користувача та адміністратора, а також схеми процесів авторизації та редагування профілю. Побудовані моделі дозволили формалізувати взаємодію між користувачами та системою ще до початку програмної реалізації, що сприяє зменшенню кількості помилок на етапі розроблення.

Окрему увагу було приділено вибору технологій реалізації програмного забезпечення. Як середовище розробки було обрано Microsoft Visual Studio, що надає широкий набір інструментів для створення, налагодження та тестування вебдодатків. Для реалізації серверної частини системи використано платформу ASP.NET Core MVC, яка забезпечує високу продуктивність, підтримку сучасних

					КвРІПЗ.220196.01.03.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис.	Дата		51

вебтехнологій та реалізацію архітектурного шаблону Model-View-Controller. Використання патерну MVC дозволило чітко розділити логіку обробки даних, представлення інформації та керування запитами користувачів, що підвищує масштабованість та підтримуваність програмного коду.

Для організації зберігання даних було обрано систему керування базами даних Microsoft SQL Server та технологію Entity Framework Core, яка забезпечує зручну взаємодію між програмним кодом і базою даних на основі об'єктно-реляційного відображення. Клієнтська частина вебресурсу реалізується за допомогою HTML5, CSS3, JavaScript та Bootstrap, що дозволяє створити сучасний адаптивний інтерфейс користувача для різних типів пристроїв.

Загалом, у результаті виконання другого розділу було сформовано повне проєктне рішення майбутнього вебсайту онлайн-галереї з продажу картин, визначено його архітектуру, змодельовано основні бізнес-процеси та обрано сучасний технологічний стек для реалізації програмного забезпечення. Отримані результати створюють основу для практичної реалізації системи та переходу до етапу програмування й тестування розробленого вебресурсу.

					КВРІПЗ.220196.01.03.ПЗ	Арк.
						52
Змін.	Арк.	№ докум.	Підпис.	Дата		

### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 3.1 Створення бази даних проекту

Одним із найважливіших етапів реалізації вебсайту онлайн-галереї з продажу картин є проектування бази даних. Саме база даних забезпечує централізоване зберігання інформації про користувачів, картини, категорії, замовлення та інші об'єкти предметної області.

Під час розроблення програмного забезпечення було використано реляційну модель даних, яка дозволяє організувати інформацію у вигляді взаємопов'язаних таблиць. Такий підхід забезпечує цілісність даних, спрощує їх обробку та дозволяє ефективно виконувати операції пошуку, додавання, редагування та видалення інформації.

Для створення бази даних застосунку нам необхідно описати всі сутності, після чого нормалізувати базу даних. Нормалізація базується на концепції нормальних форм, які є рівнями відповідності певним критеріям добре спроектованої бази даних. Чим вище нормальна форма, тим більш нормалізованою є база даних. Найпоширенішими нормальними формами є перша нормальна форма (1НФ), друга нормальна форма (2НФ) і третя нормальна форма (3НФ). Відповідно до нашого типу застосунку базу даних необхідно нормалізувати до 3НФ.

Відповідно до архітектурних вимог нашого типу застосунку (онлайн-комерція та електронні платежі), базу даних було успішно нормалізовано до 3НФ. Це дозволило уникнути дублювання інформації про клієнтів чи характеристики картин і забезпечило високу швидкість виконання транзакцій.

Для забезпечення повнофункціональної роботи та покриття всіх бізнес-процесів онлайн-галереї було спроектовано та реалізовано реляційну базу даних, яка складається з 8 взаємопов'язаних таблиць. Для повнофункціональної роботи програмного забезпечення необхідно також спроектувати та реалізувати базу

					КвРІПЗ.220196.01.03.ІЗ	Арк.
Змін.	Арк.	№ докум.	Підпис.	Дата		53

даних. В даному проекті бази даних містить 8 таблиць, опис яких подано далі (таблиця 3.1).

Таблиця 3.1 – Опис таблиць бази даних

Назва таблиці	Вміст таблиці
Users	Інформація про дані користувача.
UserRoles	Ролі користувачів
Roles	Список ролей, що можуть надаватись користувачам
Paintings	Містить інформацію про картину
Categories	Список категорій товарів
Orders	Інформація про замовлення
OrderDetails	Деталізація інформації про здійснене замовлення
Pages	Інформація про сторінки ресурсу

Таблиця Users призначена для зберігання інформації про зареєстрованих користувачів системи (Таблиця 3.2).

Таблиця 3.2 – Структура таблиці Users

Ім'я поля	Тип даних поля	Опис
Id	int	Унікальний ідентифікатор користувача
FirstName	nvarchar(100)	Ім'я користувача
LastName	nvarchar(100)	Прізвище користувача
Email	nvarchar(255)	Електронна адреса
PasswordHash	nvarchar(500)	Зашифрований пароль
PhoneNumber	nvarchar(20)	Контактний номер телефону
Address	nvarchar(255)	Адреса користувача
RegistrationDate	datetime	Дата реєстрації
IsActive	bit	Статус активності облікового запису

Таблиця містить усі необхідні персональні дані користувачів та використовується під час авторизації і оформлення замовлень.

Таблиця Roles призначена для зберігання списку ролей, які можуть бути призначені користувачам системи (Таблиця 3.3).

Таблиця 3.3 – Структура таблиці Roles

Ім'я поля	Тип даних поля	Опис
Id	int	Унікальний ідентифікатор ролі
Name	nvarchar(50)	Назва ролі
Description	nvarchar(255)	Опис ролі

Основними ролями системи є «Адміністратор» та «Користувач».

Таблиця UserRoles реалізує зв'язок між користувачами та ролями у нашій базі даних (Таблиця 3.4).

Таблиця 3.4 – Структура таблиці UserRoles

Ім'я поля	Тип даних поля	Опис
Id	int	Унікальний ідентифікатор запису
UserId	int	Ідентифікатор користувача
RoleId	int	Ідентифікатор ролі

Використання окремої таблиці дозволяє реалізувати гнучку систему керування правами доступу.

Таблиця Categories містить перелік категорій картин (Таблиця 3.5).

Таблиця 3.5 – Структура таблиці Categories

Ім'я поля	Тип даних поля	Опис
Id	int	Унікальний ідентифікатор категорії
Name	nvarchar(100)	Назва категорії
Description	nvarchar(500)	Опис категорії

Ім'я поля	Тип даних поля	Опис
ImageUrl	nvarchar(255)	Зображення категорії

Прикладами категорій можуть бути пейзажі, натюрморти, портрети, абстракція та сучасне мистецтво.

Таблиця Paintings є основною таблицею системи та містить інформацію про картини, представлені в онлайн-галереї (Таблиця 3.6).

Таблиця 3.6 – Структура таблиці Paintings

Ім'я поля	Тип даних поля	Опис
Id	int	Унікальний ідентифікатор картини
Title	nvarchar(200)	Назва картини
Artist	nvarchar(150)	Автор картини
CategoryId	int	Ідентифікатор категорії
Description	nvarchar(max)	Опис картини
Price	decimal(18,2)	Вартість картини
Width	decimal(10,2)	Ширина картини
Height	decimal(10,2)	Висота картини
Material	nvarchar(150)	Матеріал виконання
CreationYear	int	Рік створення
ImageUrl	nvarchar(255)	Посилання на зображення
Quantity	int	Кількість доступних екземплярів
IsAvailable	bit	Статус наявності

Дана таблиця використовується для формування каталогу товарів та відображення інформації на вебсайті для продажу та покупки творів художніх мистецтв, зокрема картин .

Таблиця Orders призначена для зберігання загальної інформації про замовлення (Таблиця 3.7).

					КВРППЗ.220196.01.03.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис.	Дата		56



Таблиця 3.9 – Структура таблиці Pages

Ім'я поля	Тип даних поля	Опис
Id	int	Унікальний ідентифікатор сторінки
Title	nvarchar(200)	Заголовок сторінки
Slug	nvarchar(200)	URL-адреса сторінки
Content	nvarchar(max)	Вміст сторінки
CreatedDate	datetime	Дата створення
UpdatedDate	datetime	Дата останнього оновлення
IsPublished	bit	Статус публікації

До таких сторінок можуть належати розділи «Про нас», «Доставка та оплата», «Контакти», «Політика конфіденційності» та інші інформаційні матеріали.

Спроектвана структура бази даних містить систему взаємозв'язків між таблицями.

Основними зв'язками є:

- Users (1) → (M) Orders;
- Roles (1) → (M) UserRoles;
- Users (1) → (M) UserRoles;
- Categories (1) → (M) Paintings;
- Orders (1) → (M) OrderDetails;
- Paintings (1) → (M) OrderDetails.

Завдяки використанню зовнішніх ключів забезпечується цілісність даних та коректність роботи всієї інформаційної системи.

У результаті проектування бази даних було сформовано структуру, яка забезпечує зберігання та обробку всієї інформації, необхідної для функціонування вебсайту онлайн-галереї з продажу картин. Розроблена база даних складається з восьми взаємопов'язаних таблиць, що охоплюють інформацію про користувачів, ролі, картини, категорії, замовлення та інформаційні сторінки ресурсу. Використання реляційної моделі даних дозволяє забезпечити цілісність інформації, високу продуктивність роботи системи та можливість її подальшого розвитку.



багаторівневу архітектуру програмного забезпечення та забезпечити чітке розділення відповідальностей між окремими компонентами системи.

Основними компонентами серверної частини є:

- моделі даних;
- контролери;
- сервіси бізнес-логіки;
- система авторизації;
- модуль роботи з базою даних;
- модуль обробки замовлень.

Реалізація моделей даних є основою програмного забезпечення та відображають структуру таблиць бази даних у вигляді об'єктів мови програмування C#.

Для кожної таблиці бази даних було створено відповідний клас моделі:

- User;
- Role;
- UserRole;
- Painting;
- Category;
- Order;
- OrderDetail;
- Page.

Моделі містять властивості, які відповідають полям таблиць бази даних, а також навігаційні властивості для реалізації зв'язків між сутностями.

Використання моделей дозволяє працювати з даними на рівні об'єктів без необхідності створення SQL-запитів вручну.

Реалізація контролерів є центральним елементом архітектури MVC та відповідають за обробку HTTP-запитів користувачів.

У системі було реалізовано наступні контролери:

- HomeController;

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						60
Змін.	Арк.	№ докум.	Підпис.	Дата		

- AccountController;
- PaintingController;
- CategoryController;
- CartController;
- OrderController;
- AdminController;
- PageController.

HomeController відповідає за формування головної сторінки вебсайту та відображення основної інформації.

PaintingController забезпечує роботу з каталогом картин, перегляд детальної інформації про товар та пошук необхідних позицій.

CategoryController використовується для відображення картин відповідно до вибраної категорії.

CartController реалізує механізм роботи кошика покупок.

OrderController забезпечує оформлення замовлень та їх подальшу обробку.

AccountController відповідає за реєстрацію, авторизацію та редагування профілів користувачів.

AdminController реалізує функціонал адміністративної панелі.

PageController використовується для відображення інформаційних сторінок сайту.

При реалізації авторизації та автентифікації, для забезпечення безпечного доступу до системи, було реалізовано механізм автентифікації користувачів.

Авторизація здійснюється за допомогою логіна та пароля.

Під час входу до системи виконується:

1. Перевірка введених облікових даних.
2. Пошук користувача у базі даних.
3. Перевірка зашифрованого пароля.
4. Створення користувацької сесії.
5. Надання відповідних прав доступу.

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						61
Змін.	Арк.	№ докум.	Підпис.	Дата		

Для захисту облікових записів використовується хешування паролів та механізми безпеки ASP.NET Core Identity.

У системі реалізовано два рівні доступу –користувач та адміністратор. Кожна роль має власний набір дозволених операцій.

Реалізація пошуку та фільтрації картин є однією з найважливіших функцій онлайн-галереї є можливість швидкого пошуку необхідних картин.

Для цього було реалізовано механізм пошуку за назвою картини, автором, категорією та ключовими словами.

Крім пошуку, система підтримує фільтрацію каталогу за такими параметрами як, ціна, категорія, популярність, дата додавання, наявність товару.

Завдяки цьому користувач може швидко знайти необхідний твір мистецтва навіть за великої кількості представлених картин.

Реалізація кошика покупок є одним із ключових компонентів системи електронної комерції. Його функціональні можливості включають: додавання товарів, видалення товарів, зміну кількості позицій, автоматичний розрахунок вартості замовлення та формування підсумкової суми.

Інформація про кошик зберігається протягом сеансу роботи користувача та використовується під час оформлення замовлення.

Після вибору картин користувач переходить до оформлення замовлення. Процес оформлення включає перевірку кошика, введення контактної інформації, вибір способу доставки, вибір способу оплати та підтвердження замовлення.

Після успішного оформлення система автоматично створює запис у таблиці Orders та відповідні записи у таблиці OrderDetails.

Усі замовлення відображаються в адміністративній панелі для подальшої обробки.

Для керування ресурсом була реалізована окрема адміністративна панель. Адміністратор отримує можливість додавати нові картини, редагувати інформацію про картини, видалити картини, керувати категоріями, переглядати замовлення, змінювати статуси замовлень, керувати інформаційними сторінками сайту.

					КвРІПЗ.220196.01.03.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис.	Дата		62

Використання адміністративної панелі дозволяє підтримувати актуальність контенту без внесення змін до програмного коду.

### 3.3 Реалізація клієнтської частини програмного забезпечення та інтерфейсу

Клієнтська частина вебсайту забезпечує безпосередню взаємодію користувача із системою. Основним її завданням є відображення інформації та забезпечення зручного користувацького інтерфейсу.

Розробка клієнтської частини здійснювалася із використанням сучасних вебтехнологій HTML5, CSS3, JavaScript, Bootstrap, \* Razor Views.

Використання даних технологій дозволило створити адаптивний інтерфейс, який коректно працює на різних пристроях.

Головна сторінка є основною точкою входу користувача до системи.

На ній розміщено:

- навігаційне меню;
- пошуковий рядок;
- каталог популярних картин;
- категорії товарів;
- інформаційні блоки;
- контактну інформацію.

Головна сторінка забезпечує швидкий доступ до основних функцій вебресурсу (Рисунок 3.2).

Після того, як авторизований користувач натискає кнопку “Логін” то здійснюється відкриття сторінки, де йому можна пройти авторизацію. Для цього користувачу необхідно здійснити введення даних відповідно для свого акаунту (логін і пароль). На рисунку 3.3 демонструється вигляд сторінки для здійснення авторизації.

					КвРІПЗ.220196.01.03.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис.	Дата		63





Вибір картини користувачем та добавлення її у корзину здійснюється за допомогою кнопки «Додати до кошика». Далі обрана картина переміщується у кошик покупця. Потім можна здійснити оплату, або ж продовжити здійснювати інші покупки. На рисунку 3.6 представлено загальний вигляд сторінки кошика.

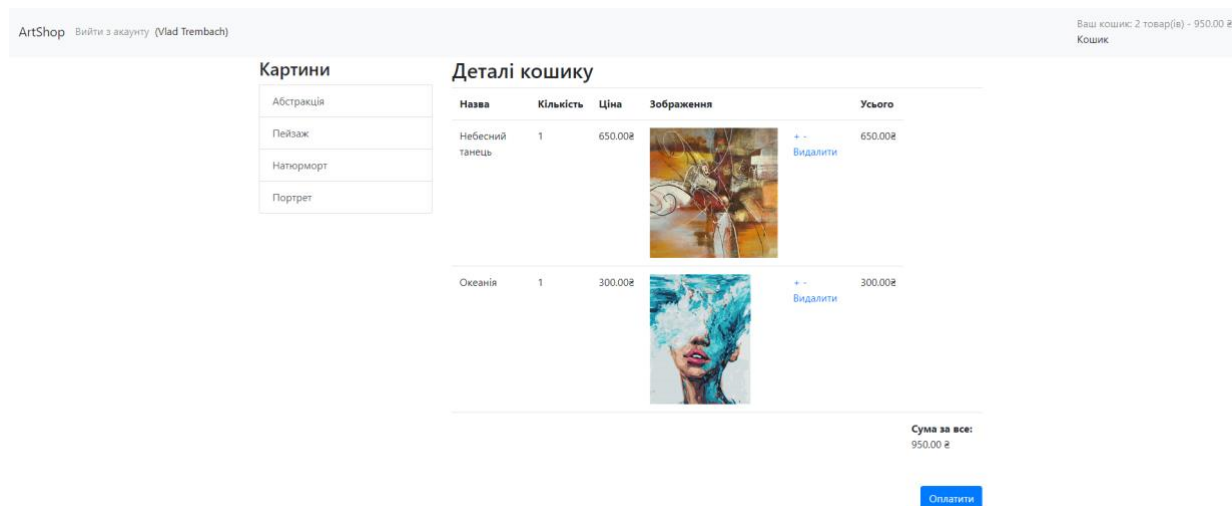


Рисунок 3.6 – Кошик покупок

Користувач має можливість змінювати склад замовлення без повернення до каталогу.

Після того, як користувач натиснув кнопку «Оплатити» здійснюється автоматичне перенаправлення його на підключення системи оплати (Рисунок 3.7)

Адміністратор системи також має свої можливості для здійснення певних дій в залежності від його потреб. Після того, коли була здійснена авторизація адміністратор ресурсу попадає на головну сторінку вебресурсу та має можливість переходу до адмінзони. Після вдалої авторизації сам адміністратор може керувати вмістим ресурсу, а також здійснювати певні корегування у разі необхідності.

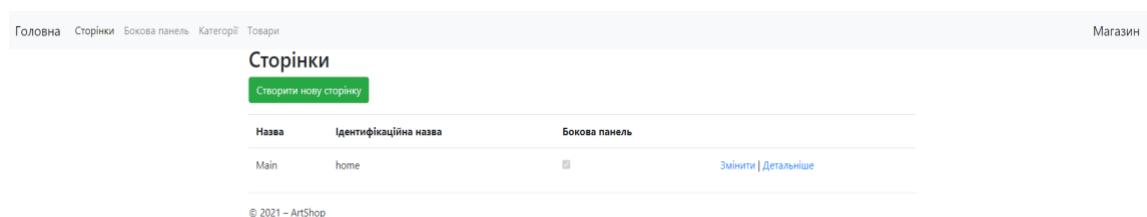


Рисунок 3.7 – Вигляд сторінки для редагування сторінок сайту

На рисунку 3. 8 наведено як виглядає спеціальний редактор, з допомогою якого здійснюється зміни вигляду головної сторінки.

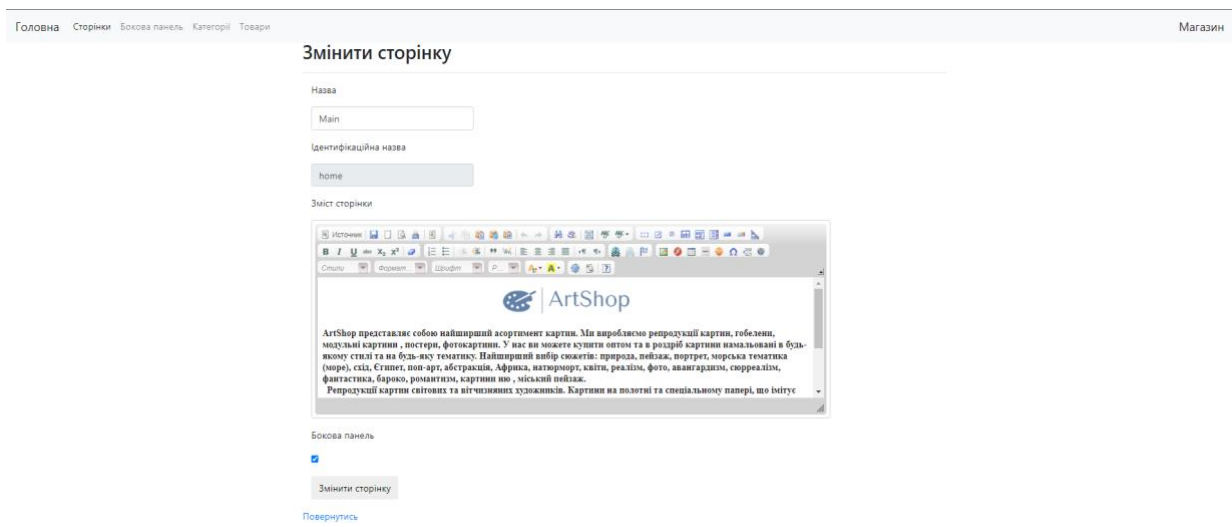


Рисунок 3.8 – Редагування сторінок адміністратором

На рисунку 3.9 показано можливості, як адміністратор здійснює додавання нових та видалення вже існуючих категорій товарів, що відображаються у каталозі.

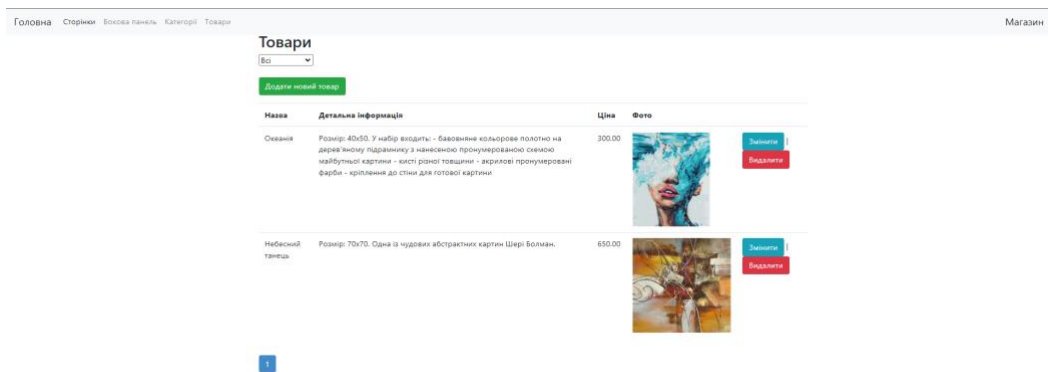


Рисунок 3.9 – Вигляд сторінки управління товарами

Однією з важливих вимог до програмного забезпечення була підтримка різних типів пристроїв.

Для реалізації адаптивності використовувався Bootstrap Framework. Завдяки цьому вебсайт коректно відображається на персональних комп'ютерах, ноутбуках, планшетах та смартфонах.

Адаптивний дизайн дозволяє забезпечити комфортну роботу користувачів незалежно від роздільної здатності екрана та типу пристрою.

### 3.4 Тестування програмного забезпечення

Після завершення розроблення вебсайту онлайн-галереї з продажу картин було проведено комплексне тестування програмного забезпечення. Основною метою тестування є перевірка коректності роботи всіх функціональних модулів системи, виявлення можливих помилок та підтвердження відповідності програмного продукту вимогам, сформованим на етапі проєктування.

Тестування є одним із найважливіших етапів життєвого циклу програмного забезпечення, оскільки дозволяє оцінити якість реалізованої системи, перевірити її надійність та забезпечити стабільну роботу в реальних умовах експлуатації.

Під час тестування вебсайту онлайн-галереї використовувалися методи функціонального та інтеграційного тестування. Особлива увага приділялася перевірці роботи механізмів авторизації, каталогу картин, оформлення замовлень, адміністративної панелі та взаємодії із базою даних.

#### 3.4.1 Тестування авторизації та реєстрації користувачів

Одним із перших етапів тестування стала перевірка механізмів реєстрації та авторизації користувачів.

Основними завданнями тестування були:

- перевірка коректності створення нового облікового запису;
- перевірка валідації введених даних;
- перевірка коректності входу до системи;
- перевірка захисту від несанкціонованого доступу.

Для перевірки реєстрації нами було виконано декілька тестових сценаріїв (Таблиця 3.9).

Результати тестування показали, що система коректно виконує перевірку введених даних та не допускає створення некоректних облікових записів.

Під час тестування авторизації перевірялися різні варіанти введення логіна та пароля (Таблиця 3.10).

					КвРІПЗ.220196.01.03.ІЗ	Арк.
						68
Змін.	Арк.	№ докум.	Підпис.	Дата		

Таблиця 3.9 – Тестування реєстрації користувача

№	Тестова дія	Очікуваний результат	Результат
1	Введення коректних даних	Створення облікового запису	Успішно
2	Введення порожнього поля Email	Повідомлення про помилку	Успішно
3	Введення некоректної електронної адреси	Повідомлення про помилку	Успішно
4	Введення короткого пароля	Відмова у реєстрації	Успішно
5	Повторна реєстрація існуючого Email	Повідомлення про існуючий акаунт	Успішно

Таблиця 3.10 – Тестування авторизації

№	Тестова дія	Очікуваний результат	Результат
1	Введення правильного логіна і пароля	Вхід до системи	Успішно
2	Введення неправильного пароля	Повідомлення про помилку	Успішно
3	Введення неіснуючого Email	Відмова у вході	Успішно
4	Вхід під роллю адміністратора	Доступ до адміністративної панелі	Успішно

Наступним етапом стало тестування каталогу картин.

Основною метою перевірки було забезпечення правильного відображення інформації про товари та коректної роботи механізмів пошуку й фільтрації.

Було протестовано:

- відображення списку картин;
- відкриття сторінки товару;
- пошук картин;
- довільний пошук;
- фільтрацію за категоріями;
- відображення інформації про автора;
- відображення ціни та опису.

Таблиця 3.11 – Тестування каталогу картин

№	Тестова дія	Очікуваний результат	Результат
1	Відкриття каталогу	Відображення всіх картин	Успішно
2	Вибір категорії	Відображення відповідних товарів	Успішно
3	Пошук за назвою	Відображення знайдених картин	Успішно
4	Перехід на сторінку товару	Відображення повної інформації	Успішно
5	Перевірка ціни товару	Коректне відображення ціни	Успішно

Під час тестування не було виявлено помилок відображення або втрати даних.

### 3.4.3 Тестування кошика покупок

Для перевірки механізму оформлення покупок було виконано тестування кошика користувача. Було перевірено такі функції:

- додавання товару до кошика;
- видалення товару;
- зміна кількості товарів;
- автоматичний перерахунок вартості замовлення.

Таблиця 3.12 – Тестування кошика покупок

№	Тестова дія	Очікуваний результат	Результат
1	Додавання картини до кошика	Товар відображається у кошику	Успішно
2	Видалення товару	Товар видаляється	Успішно
3	Зміна кількості	Перерахунок суми замовлення	Успішно
4	Очищення кошика	Кошик стає порожнім	Успішно

Усі операції виконувалися без помилок та коректно взаємодіяли з базою даних.

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						70
Змін.	Арк.	№ докум.	Підпис.	Дата		

### 3.4.4 Тестування оформлення замовлень

Однією з найважливіших функцій системи є оформлення замовлень. Метою тестування було підтвердження правильності формування замовлення та його збереження у базі даних. Було виконано перевірку наступних сценаріїв:

- оформлення замовлення авторизованим користувачем;
- перевірка запису в таблиці Orders;
- перевірка запису в таблиці OrderDetails;
- перевірка розрахунку вартості замовлення.

Таблиця 3.13 – Тестування оформлення замовлень

№	Тестова дія	Очікуваний результат	Результат
1	Створення замовлення	Замовлення збережено	Успішно
2	Перевірка складу замовлення	Дані відповідають кошику	Успішно
3	Перевірка суми замовлення	Коректний розрахунок	Успішно
4	Перевірка статусу замовлення	Статус «Нове»	Успішно

Після завершення тестування було підтверджено правильність роботи механізму оформлення замовлень.

### 3.4.5 Тестування адміністративної панелі

Для забезпечення коректного керування вебресурсом було проведено тестування адміністративної панелі.

Перевірялися наступні функції:

- додавання нових картин;
- редагування картин;
- видалення картин;
- керування категоріями;

- перегляд замовлень;
- редагування інформаційних сторінок.

Таблиця 3.14 – Тестування адміністративної панелі

№	Тестова дія	Очікуваний результат	Результат
1	Додавання нової картини	Створення запису в базі даних	Успішно
2	Редагування картини	Збереження змін	Успішно
3	Видалення картини	Видалення запису	Успішно
4	Створення категорії	Нова категорія створена	Успішно
5	Перегляд замовлень	Відображення списку замовлень	Успішно

Тестування показало стабільну роботу всіх адміністративних функцій системи.

### 3.4.6 Тестування продуктивності та адаптивності

Окрему увагу було приділено оцінці продуктивності програмного забезпечення.

Під час тестування перевірялися:

- швидкість відкриття сторінок;
- швидкість пошуку картин;
- швидкість оформлення замовлення;
- робота на різних пристроях.

Результати показали, що середній час завантаження сторінок становив менше 3 секунд, що відповідає встановленим нефункціональним вимогам.

Також було виконано перевірку адаптивності інтерфейсу на різних пристроях:

- персональний комп'ютер;
- ноутбук;

					КвРІПЗ.220196.01.03.ІЗ	Арк.
Змін.	Арк.	№ докум.	Підпис.	Дата		72

- планшет;
- смартфон.

Усі сторінки коректно відображалися незалежно від розміру екрана та роздільної здатності пристрою.

За результатами проведеного тестування було встановлено, що всі основні функціональні модулі системи працюють відповідно до вимог технічного завдання. Було підтверджено коректність роботи:

- механізму реєстрації;
- авторизації користувачів;
- каталогу картин;
- пошуку та фільтрації;
- кошика покупок;
- оформлення замовлень;
- адміністративної панелі;
- бази даних.

Критичних помилок, які б унеможливили використання програмного продукту, виявлено не було.

### 3.5 Висновки третього розділу

У результаті виконання програмної реалізації було створено серверну та клієнтську частини вебсайту онлайн-галереї з продажу картин. Серверна частина забезпечує обробку запитів, роботу з базою даних, авторизацію користувачів та керування замовленнями. Клієнтська частина реалізує сучасний адаптивний інтерфейс, який забезпечує зручну взаємодію користувачів із вебресурсом. Розроблене програмне забезпечення повністю відповідає вимогам, сформованим на етапі проектування, та забезпечує ефективну роботу системи електронної комерції у сфері продажу творів мистецтва.

У результаті проведеного тестування підтверджено працездатність вебсайту онлайн-галереї з продажу картин та відповідність його функціональних

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						73
Змін.	Арк.	№ докум.	Підпис.	Дата		

можливостей поставленим вимогам. Перевірка основних модулів системи показала стабільну роботу механізмів авторизації, пошуку, оформлення замовлень та адміністрування ресурсу. Результати тестування свідчать про готовність програмного продукту до практичного використання та подальшого впровадження.

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						74
Змін.	Арк.	№ докум.	Підпис.	Дата		

## ВИСНОВКИ

У результаті виконання дипломної роботи було розроблено вебсайт онлайн-галереї з продажу картин, який забезпечує автоматизацію процесів представлення, пошуку та реалізації творів мистецтва через мережу Інтернет. Актуальність теми дослідження обумовлена стрімким розвитком електронної комерції та зростанням попиту на сучасні цифрові платформи для продажу художніх робіт. Використання вебтехнологій дозволяє значно розширити аудиторію потенційних покупців, забезпечити доступність мистецьких творів незалежно від географічного розташування користувача та спростити процес взаємодії між продавцем і покупцем.

У першому розділі дипломної роботи було проведено дослідження предметної області та виконано постановку задачі. Проаналізовано особливості функціонування сучасних вебресурсів електронної комерції, розглянуто специфіку онлайн-продажу картин та визначено основні вимоги до майбутнього програмного продукту. Було досліджено існуючі аналоги вебсайтів для продажу мистецьких творів, зокрема ArtSale.ua, Art-Holst та Jose Art Gallery. На основі проведеного аналізу встановлено переваги та недоліки існуючих рішень, що дозволило сформулювати перелік функціональних та нефункціональних вимог до розроблюваної системи. Також було визначено основні категорії користувачів вебресурсу, їх функціональні можливості та сформульовано завдання проєктування програмного забезпечення.

У другому розділі виконано проєктування програмного забезпечення та обґрунтовано вибір методів і засобів його реалізації. Для побудови системи було обрано клієнт-серверну архітектуру, яка забезпечує централізоване зберігання даних, високу продуктивність та можливість масштабування програмного продукту. Було побудовано UML-діаграми варіантів використання, діаграми станів та діаграми послідовності, що дозволило формалізувати логіку функціонування системи та взаємодію користувачів із вебресурсом. Також було виконано аналіз

					КвРІПЗ.220196.01.03.ІЗ	Арк.
						75
Змін.	Арк.	№ докум.	Підпис.	Дата		

сучасних технологій веброзробки та обрано технологічний стек для реалізації проєкту. Як платформу розробки використано ASP.NET Core MVC, середовище програмування Microsoft Visual Studio, систему керування базами даних Microsoft SQL Server та технологію Entity Framework Core. Для створення клієнтської частини використано HTML5, CSS3, JavaScript та Bootstrap.

У третьому розділі було виконано практичну реалізацію програмного забезпечення. Розроблено структуру бази даних, яка складається з таблиць користувачів, ролей, картин, категорій, замовлень, деталізації замовлень та інформаційних сторінок ресурсу. Побудована база даних забезпечує цілісність інформації та підтримує всі бізнес-процеси вебсайту. Реалізовано серверну частину системи на основі архітектурного шаблону MVC, що дозволило розділити бізнес-логіку, інтерфейс користувача та механізми обробки запитів. Було створено функціональні модулі для реєстрації та авторизації користувачів, роботи з каталогом картин, пошуку та фільтрації товарів, формування кошика покупок, оформлення замовлень та адміністрування вебресурсу.

Особливу увагу було приділено розробці клієнтської частини системи. Створено адаптивний інтерфейс користувача, який забезпечує коректне відображення інформації на персональних комп'ютерах, планшетах та мобільних пристроях. Реалізовано головну сторінку ресурсу, каталог картин, сторінки товарів, форми авторизації та реєстрації, кошик покупок та адміністративну панель керування. Завдяки використанню сучасних вебтехнологій забезпечено зручність навігації та високий рівень користувацького досвіду.

На завершальному етапі роботи було проведено комплексне тестування програмного забезпечення. Перевірено працездатність усіх основних функціональних модулів системи, включаючи механізми реєстрації та авторизації користувачів, каталог картин, кошик покупок, оформлення замовлень та адміністративну панель. Результати тестування підтвердили коректність функціонування програмного продукту та його відповідність встановленим вимогам. Критичних помилок, які б перешкождали використанню системи, виявлено не було.

					КвРІПЗ.220196.01.03.ПЗ	Арк.
						76
Змін.	Арк.	№ докум.	Підпис.	Дата		

У результаті виконання дипломної роботи поставлена мета була досягнута, а всі визначені завдання успішно виконані. Розроблений вебсайт онлайн-галереї забезпечує ефективну організацію процесу продажу картин через мережу Інтернет, дозволяє автоматизувати обробку замовлень, спрощує взаємодію між покупцями та адміністрацією ресурсу, а також створює сучасне середовище для представлення та реалізації творів мистецтва.

Практична цінність отриманих результатів полягає у можливості використання розробленого програмного забезпечення як готового рішення для організації онлайн-продажу картин та інших мистецьких виробів. Розроблена система має гнучку архітектуру та може бути розширена шляхом додавання нових функціональних можливостей, інтеграції із зовнішніми платіжними сервісами, системами доставки, рекомендаційними механізмами та засобами аналітики. Це забезпечує перспективи подальшого розвитку проєкту та його адаптації до потреб сучасного ринку електронної комерції.

					КВРІПЗ.220196.01.03.ПЗ	Арк.
						77
Змін.	Арк.	№ докум.	Підпис.	Дата		

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Радельчук Г. І. Наскрізна практична підготовка: програма та методичні вказівки щодо її виконання студентами спеціальності 121 «Інженерія програмного забезпечення» освітнього ступеня «бакалавр» / Г. Радельчук, Л. П. Бедратюк. – Хмельницький, 2021. – 40 с.
2. ASP.NET Core 2.0 MVC & Razor Pages for Beginners: How to Build a
3. Website. Д. Фагерберг. Microsoft Press 2021. – 172с.
4. ASP.NET Core in Action. Автор Е. Лок. Рік видання – 2021. – 53с.
5. Pro Entity Framework Core 2 for ASP.NET Core MVC. А. Фрімен. Microsoft Press. 2018. – 17с.
6. Програмування баз даних Microsoft SQL Server 2023 для професіоналів/ Вієйра Р.. Діалектика, 2022. – 1072 с
7. Бондарев, В.М. «Об'єктно-орієнтовне програмування на С#»: навч.посіб. / В.М. Бондарев.- Х.:Компанія СМІТ, 2009.- 224 с.
8. ASP.NET Режим доступу: URL:<http://www.asp.net/mvc> - 10.04.2026 р.
9. Бібліотека розробника MSDN [Електронний ресурс] / портал [msdn.microsoft.com](http://msdn.microsoft.com) : Microsoft Developer Network. - Режим доступу: URL: <http://msdn.microsoft.com/library/>
10. Мак-Дональд, М. Microsoft ASP.NET 3.5 з прикладами на С# 2018. CLR via C# / Jeffrey Richter. Microsoft Press, 2012. - 896 с.
11. SQL: повний посібник [Текст] / . Грофф, Джеймс, Вайнберг Пол. Київ: BHV, 2025. - 608 с.
12. Myers, G. J. The Art of SoftwareTesting [Текст] / G. J. Myers. - John Wiley & Sons, Inc., Hoboken, New Jersey, Canada, 2004. - 255 с.
13. C# Coding Conventions [Електронний ресурс].– Режим доступу до ресурсу: URL: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>.

					КВРІПЗ.220196.01.03.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис.	Дата		78

14. MVC Framework – Introduction. URL: [https://www.tutorialspoint.com/mvc\\_framework/mvc\\_framework\\_introduction.htm](https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm) (дата звернення 19.02.2026)

15. Rick Anderson Getting started with ASP.NET MVC 5 [Електронний ресурс]. // Сайт <https://docs.microsoft.com/> / Rick Anderson. – Дата розміщення: 19.02.2026. Режим доступу: URL: <https://docs.microsoft.com/ua-ua/aspnet/mvc/overview/getting->

16. Introduction to HTML [Електронний ресурс]. Режим доступу: URL: [https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction\\_to\\_HTML](https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML) (дата звернення 11.02.2026)

17. Global strategy on digital health 2020-2025. Geneva: World Health Organization; 2021. Ліцензія: CC BY-NC-SA 3.0 IGO (дата звернення: 22.03.2026).

18. Don't Be Tempted to Use Expired Medicines. *U.S. Food and Drug Administration (FDA)*. URL: <https://www.fda.gov/drugs/special-features/dont-be-tempted-use-expired-medicines> (дата звернення: 22.03.2026).

19. UHBlog. Expired Medications: Dangerous or Just Less Effective?. *Nationally Ranked Healthcare - Largest Network of Hospitals, Doctors & Surgeons in Cleveland & Northeast Ohio | University Hospitals*. URL: <https://www.uhhospitals.org/blog/articles/2023/08/expired-medications-dangerous-or-just-less-effective> (дата звернення: 22.03.2026).

20. Scheduling a notification locally from your app | Apple Developer Documentation. *Apple Developer Documentation*. URL: <https://developer.apple.com/documentation/usernotifications/scheduling-a-notification-locally-from-your-app> (дата звернення: 22.03.2026).

21. Color | Apple Developer Documentation. *Apple Developer Documentation*. URL: <https://developer.apple.com/design/human-interface-guidelines/color> (дата звернення: 22.03.2026).

22. Andrzejewska M., Stolińska A., Baran W. The Effects of Colour Coding on Problem-Solving Strategies and Cognitive Engagement: Insights from Eye-Tracking

						КВРІПЗ.220196.01.03.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис.	Дата			79

Research. *Applied Sciences*. 2025. Т. 15, № 21. С. 11503. URL: <https://doi.org/10.3390/app152111503> (дата звернення: 22.03.2026).

23. What is Deep Linking: Deep Linking Definition | Unity. *Unity*. URL: <https://unity.com/glossary/deep-linking> (дата звернення: 22.02.2026).

24. About Us. *Medisafe Inc*. URL: <https://www.medisafe.com/about-us/> (дата звернення: 22.03.2026).

25. Medication Reminder and Pill Tracker App – MyTherapy. *Medication Reminder and Pill Tracker App – MyTherapy*. URL: <https://www.mytherapyapp.com/> (дата звернення: 22.03.2026).

26. Medisafe Medication Management app App Store. *App Store*. URL: <https://apps.apple.com/ua/app/medisafe-medication-management/id573916946?l=uk> (дата звернення: 22.03.2026).

27. Нагадування таблетки MyTherapy. App Store. URL: <https://apps.apple.com/ua/app/нагадування-таблетки-mytherapy/id662170995?l=uk>

28. Home Medkit - Smart Pill & Reminder App. *Home Medkit - Smart Pill & Reminder App*. URL: <https://medkit.space/#/> (дата звернення: 22.03.2026).

29. MedKit: MedicationTracker. App Store. URL: <https://apps.apple.com/ua/app/medkit-medication-tracker/id6742353036?l=uk> (дата звернення: 22.03.2026).

30. VisionKit | Apple Developer Documentation. *Apple Developer Documentation*. URL: <https://developer.apple.com/documentation/visionkit> (дата звернення: 14.04.2026).

31. Human Interface Guidelines | Apple Developer Documentation. *Apple Developer Documentation*. URL: <https://developer.apple.com/design/human-interface-guidelines> (дата звернення: 14.04.2026).

32. Model-View-ViewModel (MVVM) : Apple Developer Documentation. URL: <https://developer.apple.com/documentation/swiftui/managing-model-data-in-your-app> (дата звернення: 19.04.2026).

					КВРІПЗ.220196.01.03.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис.	Дата		80

33. Учасники проєктів Вікімедіа. Model-View-ViewModel – Вікіпедія. *Вікіпедія*. URL: <https://uk.wikipedia.org/wiki/Model-View-ViewModel>(дата звернення: 19.04.2026).

34. Indrawan D., Kusumo D. S., Puspitasari S. Y. ANALYSIS OF THE IMPLEMENTATION OF MVVM ARCHITECTURE PATTERN ON PERFORMANCE OF IOS MOBILE-BASED APPLICATIONS. *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*. 2023. Vol. 8, no. 1. P. 59–65. URL: <https://doi.org/10.29100/jupi.v8i1.3293> (дата звернення: 19.03.2026).

35. Lano K., Yassipour Tehrani S. Introduction to Clean Architecture Concepts. *Introduction to Software Architecture*. Cham, 2023. С. 35–49. URL: [https://doi.org/10.1007/978-3-031-44143-1\\_2](https://doi.org/10.1007/978-3-031-44143-1_2) (дата звернення: 20.03.2026).

36. Декомпозиція в програмуванні: що це і навіщо потрібно. *FoxmindEd*. URL: <https://foxminded.ua/dekompozytsiia-v-prohramuvanni/>(дата звернення: 29.03.2026).

37. AVFoundation | Apple Developer Documentation. *Apple Developer Documentation*. URL: <https://developer.apple.com/documentation/avfoundation>(дата звернення: 24.03.2026).

38. Vision | Apple Developer Documentation. *Apple Developer Documentation*. URL: <https://developer.apple.com/documentation/vision> (дата звернення: 22.02.2026).

39. AVCaptureMetadataOutput | Apple Developer Documentation. *Apple Developer Documentation*. URL: <https://developer.apple.com/documentation/avfoundation/avcapturemetadataoutput> (дата звернення: 29.03.2026).

40. SwiftData | Apple Developer Documentation. *Apple Developer Documentation*. URL: <https://developer.apple.com/documentation/swiftdata>(дата звернення: 29.03.2026).

41. Flutter documentation. *Flutter documentation*. URL: <https://docs.flutter.dev/> (дата звернення: 14.03.2026).

42. Swift.org. *Swift.org*. URL: <https://www.swift.org/> (date of access: 24.04.2026).

						КВРІПЗ.220196.01.03.ПЗ	Арк.
							81
Змін.	Арк.	№ докум.	Підпис.	Дата			

43. SwiftUI - Apple Developer. *Apple Developer*. URL: <https://developer.apple.com/swiftui/> (дата звернення: 24.04.2026).

44. Xcode - Apple Developer. *Apple Developer*. URL: <https://developer.apple.com/xcode/> (дата звернення: 24.04.2026).

45. JetBrains. AppCode: Smart Swift/Objective-C IDE for iOS & macOS Development. *JetBrains*. URL: <https://www.jetbrains.com/objc/> (дата звернення: 24.04.2026).

46. Figma: The Collaborative Interface Design Tool. *Figma*. URL: <https://www.figma.com> (дата звернення: 21.04.2026).

47. Agile VS Waterfall – який варіант відповідаєвашому бізнесу?. *Worksection*. URL: <https://worksection.com/ua/blog/waterfall-vs-agile.html> (дата звернення: 21.04.2026).

48. Ітеративна модель (iterative model) - QALight. *QALight*. URL: <https://qalight.ua/baza-znaniy/iterativna-model-iterative-model/> (дата звернення: 15.05.2026).

					КВРІПЗ.220196.01.03.ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис.	Дата		82

ДОДАТОК А  
(обов'язковий)

**ТЕХНІЧНЕ ЗАВДАННЯ**

## **Введення**

Робота виконується в рамках проєкту «Вебсайт онлайн-галереї з продажу картин».

### **1 Підстава для розробки**

Підставою для розробки є «Завдання на кваліфікаційну роботу», затверджене завідувачем кафедри програмного забезпечення. Найменування розробки: «Вебсайт онлайн-галереї з продажу картин».

### **2 Призначення розробки**

Веб-застосунок онлайн-галереї з продажу картин призначений для автоматизації процесів реалізації картин (рисуноків, малюнків, ескізів тощо та обліку продаж. Користувачами програми є кінцеві покупці, менеджери з продажу та адміністратори системи.

Застосунок дозволяє клієнтам переглядати актуальний каталог картин, додавати їх до інтерактивного кошика та оформлювати замовлення онлайн. Система забезпечує зберігання історії покупок, керування особистими даними користувача та надання зворотного зв'язку щодо сервісного обслуговування.

### **3 Вимоги до програми**

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

Веб-застосунок онлайн-галереї з продажу картин повинен забезпечувати виконання наступних функцій:

- ведення та динамічне оновлення каталогу картин за категоріями;
- реєстрація, авторизація та управління профілями користувачів;
- механізм фільтрації та швидкого пошуку картин за характеристиками;
- управління вмістом кошика та автоматичний розрахунок загальної вартості;
- формування та зберігання електронних замовлень у базі даних SQL Server;
- адміністративний інтерфейс для модерації картин та перегляду статистики продажів.

#### **3.2 Вимоги до надійності**

Розроблюване ПЗ повинно мати:

- захист персональних даних користувачів за допомогою шифрування паролів;
- валідацію вхідних даних для запобігання некоректним записам у БД;
- стійкість до збоїв на стороні сервера та можливість швидкого відновлення сесії;
- обмеження доступу до адміністративних функцій через систему ролей.

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

Системні вимоги для роботи ПЗ:

– серверна частина: тактова частота процесора від 2.0 ГГц, обсяг ОЗП від 1 Гб, вільний простір на диску від 200 Мб;

– клієнтська частина: будь-який сучасний пристрій з доступом до мережі Інтернет та встановленим веб-браузером (Chrome, Firefox, Safari).

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

Програма повинна бути реалізована на платформі .NET з використанням бібліотеки React. Серверна логіка базується на ASP.NET Core, зберігання даних здійснюється у Microsoft SQL Server. Веб-застосунок має бути сумісним з сучасними операційними системами (Windows, Android, iOS, macOS).

### **3.5 Спеціальні вимоги**

Програма повинна мати адаптивний інтерфейс (Responsive Design), розрахований на коректне відображення на мобільних пристроях та десктопних моніторах. Архітектура системи повинна передбачати можливість масштабування та інтеграції з платіжними системами.

## **4 Вимоги до програмної документації**

В ході розробки програми повинні бути підготовлені: текст програми, опис архітектури, результати тестування, керівництво користувача та адміністратора.

## **5 Техніко-економічне обґрунтування**

Зараз обробка замовлень у магазині відбувається переважно в ручному режимі (через месенджери або телефон), що займає до 15-20 хвилин на одного клієнта. Впровадження системи онлайн-галереї з продажу картин дозволяє скоротити час оформлення замовлення до 2-3 хвилин. Автоматичне формування звітів про продажі за місяць замінює 4-6 годин ручної роботи бухгалтера на 30 секунд автоматичного запиту до БД. Виключення людського фактора при розрахунку вартості товарів та залишків на складі мінімізує фінансові ризики підприємства.

## **6 Стадії та етапи розробки**

Таблиця 1. Стадії та етапи розробки

Стадія розробки	Етапи робіт	Зміст робіт
Технічне завдання (05.01 – 30.01.2026)	Обґрунтування необхідності розробки програми	Аналіз ринку е-комерції велотоварів; опис призначення системи Velo-hub; формування вимог до ПЗ та документації; стадії і етапи розробки.
Ескізний проект (01.02 – 15.02.2026)	Розробка ескізного проекту	Визначення структури вхідних (каталог) та вихідних даних (замовлення); вибір стеку технологій (.NET, React, SQL Server); опис загальної архітектури.

Технічний проєкт (16.02 – 05.03.2026)	Розробка технічного проєкту	Проектування схеми бази даних (таблиці Users, Products, Orders); розробка алгоритмів взаємодії фронтенду з API; визначення параметрів сервера.
Робочий проєкт (06.03 – 20.04.2026)	Розробка програмного забезпечення	Написання коду на C# та TypeScript; реалізація логіки кошика та авторизації; верстка інтерфейсу; відладка та попереднє тестування.
Розробка програмної документації (21.04 – 29.04.2026)	Складання пояснювальної записки	Опис модулів програми; підготовка керівництва користувача та адміністратора; оформлення пояснювальної записки згідно з вимогами ДСТУ.
Тестування системи (01.05 – 12.05.2026)	Проведення випробувань ПЗ	Функціональне тестування «чорною скринькою»; перевірка адаптивності верстки; аналіз швидкодії мережевих запитів (200 ОК); виправлення дефектів.
Впровадження (012.05 – 15.06.2026)	Підготовка до захисту	Розгортання системи; фінальне налаштування бази даних; демонстрація результатів роботи комісії; підписання акту приймання-передачі.

### **7 Порядок контролю та приймання**

Здійснення контролів відбувається безпосередньо користувачем застосунку підключеним на етапі тестування. Фінальне приймання програмного продукту проводиться за умови його повного розгортання в робочому середовищі, успішного завершення всіх етапів тестування та виконання фінального налагодження параметрів системи

ДОДАТОК Б  
(обов'язковий)

ПРОГРАМНИЙ КОД ОСНОВНИХ МОДУЛІВ

1 Фрагмент програмного коду роботи з базою даних

```
import tkinter as tk
from tkinter import ttk, messagebox
import mysql.connector
from mysql.connector import Error

DB_CONFIG = {
    "host": "localhost",
    "user": "root",
    "password": "nazikk",
    "database": "shop_db"
}

PLACEHOLDER = "Введіть текст для пошуку..."

class ShopApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Магазин побутової техніки")
        self.root.geometry("1200x700")
        self.root.minsize(1050, 620)
        self.root.configure(bg="#eef3f8")
        self.current = "products"

        self.queries = {
            "products": (
                "Товари магазину",
                """
                SELECT p.product_id AS ID, p.product_name AS Назва, c.category_name
                AS Категорія,
                           p.brand AS Бренд, p.model AS Модель, p.price AS Ціна,
                           p.quantity_in_stock AS Залишок, p.warranty_months AS Гарантія
                FROM products p
                JOIN categories c ON p.category_id = c.category_id
                ORDER BY p.product_id
                """,
            ),
            "orders": (
                "Замовлення клієнтів",
                """
                SELECT o.order_id AS ID, CONCAT(c.first_name, ' ', c.last_name) AS
                Клієнт,
                           CONCAT(e.first_name, ' ', e.last_name) AS Працівник,
                           o.order_date AS Дата, o.status AS Статус, o.total_amount AS
                Сума
                FROM orders o
                JOIN customers c ON o.customer_id = c.customer_id
                JOIN employees e ON o.employee_id = e.employee_id
                ORDER BY o.order_date DESC
                """,
            )
        }
```

```

),
"customers": (
    "Клієнти магазину",
    """"
        SELECT customer_id AS ID, first_name AS "Ім'я", last_name AS
Прізвище,
                phone AS Телефон, email AS Email, address AS Адреса
FROM customers
ORDER BY customer_id
    """"
),
"employees": (
    "Працівники магазину",
    """"
        SELECT employee_id AS ID, first_name AS "Ім'я", last_name AS
Прізвище,
                position AS Посада, phone AS Телефон, salary AS Зарплата
FROM employees
ORDER BY employee_id
    """"
),
"suppliers": (
    "Постачальники",
    """"
        SELECT supplier_id AS ID, supplier_name AS Постачальник,
                phone AS Телефон, email AS Email, address AS Адреса
FROM suppliers
ORDER BY supplier_id
    """"
),
"payments": (
    "Оплати",
    """"
        SELECT p.payment_id AS ID, p.order_id AS Замовлення,
                CONCAT(c.first_name, ' ', c.last_name) AS Клієнт,
                p.payment_date AS Дата, p.amount AS Сума, p.payment_method
AS Спосіб
        FROM payments p
        JOIN orders o ON p.order_id = o.order_id
        JOIN customers c ON o.customer_id = c.customer_id
        ORDER BY p.payment_date DESC
    """"
),
"deliveries": (
    "Поставки",
    """"
        SELECT d.delivery_id AS ID, s.supplier_name AS Постачальник,
                d.delivery_date AS Дата, CONCAT(e.first_name, ' ', e.last_name)
AS Працівник
        FROM deliveries d
        JOIN suppliers s ON d.supplier_id = s.supplier_id
        JOIN employees e ON d.employee_id = e.employee_id
        ORDER BY d.delivery_date DESC
    """"
),
"low_stock": (
    "Товари з малим залишком",
    """"
        SELECT product_id AS ID, product_name AS Назва, brand AS Бренд,
                model AS Модель, quantity_in_stock AS Залишок

```

```

        FROM products
        WHERE quantity_in_stock < 10
        ORDER BY quantity_in_stock
        """"
    )
}

self.make_style()
self.make_ui()
self.show("products")

def connect(self):
    try:
        return mysql.connector.connect(**DB_CONFIG)
    except Error as e:
        messagebox.showerror("Помилка підключення", str(e))
        return None

def make_style(self):
    style = ttk.Style()
    style.theme_use("clam")
    style.configure("Treeview", font=("Segoe UI", 10), rowheight=30,
                    background="white", fieldbackground="white")
    style.configure("Treeview.Heading", font=("Segoe UI", 10, "bold"),
                    background="#1f4e79", foreground="white")
    style.map("Treeview", background=[("selected", "#2563eb")],
              foreground=[("selected", "white")])

def button(self, parent, text, command, color="#2563eb"):
    return tk.Button(parent, text=text, command=command, bg=color, fg="white",
                     activebackground=color, activeforeground="white",
                     font=("Segoe UI", 10, "bold"), bd=0, padx=12, pady=8,
                     cursor="hand2")

def make_ui(self):
    tk.Label(self.root, text="Інформаційна система магазину побутової техніки",
             bg="#1f4e79", fg="white", font=("Segoe UI", 20, "bold"),
             pady=18).pack(fill="x")

    main = tk.Frame(self.root, bg="#eef3f8")
    main.pack(fill="both", expand=True, padx=15, pady=15)

    menu = tk.Frame(main, bg="#eef3f8")
    menu.pack(fill="x", pady=(0, 10))

    items = [
        ("Товари", "products"), ("Замовлення", "orders"),
        ("Клієнти", "customers"), ("Працівники", "employees"),
        ("Постачальники", "suppliers"), ("Оплати", "payments"),
        ("Поставки", "deliveries"), ("Малий залишок", "low_stock")
    ]

    for i, (text, key) in enumerate(items):
        self.button(menu, text, lambda k=key: self.show(k)).grid(
            row=i // 4, column=i % 4, padx=5, pady=5, sticky="ew"
        )

        self.button(menu, "Додати клієнта", self.add_customer_window,
                    "#16a34a").grid(
            row=2, column=0, padx=5, pady=5, sticky="ew"

```

```

    )
    self.button(menu, "Створити замовлення", self.order_window, "#16a34a").grid(
        row=2, column=1, padx=5, pady=5, sticky="ew"
    )

    for i in range(4):
        menu.grid_columnconfigure(i, weight=1)

        search = tk.Frame(main, bg="white", highlightbackground="#d1d5db",
highlightthickness=1)
        search.pack(fill="x", pady=(0, 10), ipady=8)

        tk.Label(search, text="Пошук:", bg="white", font=("Segoe UI", 11,
"bold")).pack(
            side="left", padx=12
        )

        self.search = tk.Entry(search, font=("Segoe UI", 12), bg="#f9fafb",
                                fg="#6b7280", relief="flat", width=45)
        self.search.pack(side="left", ipady=7, padx=5)
        self.search.insert(0, PLACEHOLDER)
        self.search.bind("<FocusIn>", self.clear_placeholder)
        self.search.bind("<FocusOut>", self.set_placeholder)
        self.search.bind("<Return>", lambda e: self.search_data())

        self.button(search, "Шукати", self.search_data, "#16a34a").pack(side="left",
padx=8)
        self.button(search, "Очистити", self.clear_search,
"#6b7280").pack(side="left")

        box = tk.Frame(main, bg="white", highlightbackground="#d1d5db",
highlightthickness=1)
        box.pack(fill="both", expand=True)

        self.title = tk.Label(box, text="", bg="white", fg="#1f2937",
                                font=("Segoe UI", 14, "bold"))
        self.title.pack(anchor="w", padx=15, pady=10)

        table = tk.Frame(box, bg="white")
        table.pack(fill="both", expand=True, padx=15, pady=(0, 15))

        self.tree = ttk.Treeview(table, show="headings")
        y = ttk.Scrollbar(table, orient="vertical", command=self.tree.yview)
        x = ttk.Scrollbar(table, orient="horizontal", command=self.tree.xview)

        self.tree.configure(yscrollcommand=y.set, xscrollcommand=x.set)
        self.tree.grid(row=0, column=0, sticky="nsew")
        y.grid(row=0, column=1, sticky="ns")
        x.grid(row=1, column=0, sticky="ew")

        table.grid_rowconfigure(0, weight=1)
        table.grid_columnconfigure(0, weight=1)

def clear_placeholder(self, event=None):
    if self.search.get() == PLACEHOLDER:
        self.search.delete(0, tk.END)
        self.search.config(fg="#111827")

def set_placeholder(self, event=None):
    if not self.search.get().strip():

```

```

        self.search.insert(0, PLACEHOLDER)
        self.search.config(fg="#6b7280")

def query(self, sql, params=(), fetch=True):
    con = self.connect()
    if not con:
        return [], []

    try:
        cur = con.cursor()
        cur.execute(sql, params)
        if fetch:
            rows = cur.fetchall()
            cols = [d[0] for d in cur.description]
            return cols, rows
        con.commit()
        return [], []
    except Error as e:
        con.rollback()
        messagebox.showerror("Помилка SQL", str(e))
        return [], []
    finally:
        con.close()

def show(self, key):
    self.current = key
    title, sql = self.queries[key]
    cols, rows = self.query(sql)
    self.fill(title, cols, rows)

def fill(self, title, cols, rows):
    self.title.config(text=title)
    self.tree.delete(*self.tree.get_children())
    self.tree["columns"] = cols

    for col in cols:
        self.tree.heading(col, text=col)
        width = 90 if col in ("ID", "Залишок") else 230 if col in ("Назва",
"Клієнт", "Адреса", "Постачальник") else 140
        self.tree.column(col, width=width, anchor="center")

    self.tree.tag_configure("even", background="white")
    self.tree.tag_configure("odd", background="#f3f6fb")

    for i, row in enumerate(rows):
        self.tree.insert("", "end", values=row, tags=("even" if i % 2 == 0 else
"odd",))

def clear_search(self):
    self.search.delete(0, tk.END)
    self.set_placeholder()
    self.show(self.current)

def search_data(self):
    text = self.search.get().strip()

    if not text or text == PLACEHOLDER:
        messagebox.showwarning("Пошук", "Введіть текст для пошуку")
        return

```

```

like = f"%{text}%"

search_sql = {
    "products": ""
        SELECT p.product_id AS ID, p.product_name AS Назва, c.category_name
AS Категорія,
                p.brand AS Бренд, p.model AS Модель, p.price AS Ціна,
                p.quantity_in_stock AS Залишок
        FROM products p
        JOIN categories c ON p.category_id = c.category_id
        WHERE p.product_name LIKE %s OR p.brand LIKE %s OR p.model LIKE %s
OR c.category_name LIKE %s
        ORDER BY p.product_name
    "",
    "low_stock": ""
        SELECT product_id AS ID, product_name AS Назва, brand AS Бренд,
                model AS Модель, quantity_in_stock AS Залишок
        FROM products
        WHERE quantity_in_stock < 10 AND
                (product_name LIKE %s OR brand LIKE %s OR model LIKE %s OR
CAST(product_id AS CHAR) LIKE %s)
        ORDER BY quantity_in_stock
    "",
    "orders": ""
        SELECT o.order_id AS ID, CONCAT(c.first_name, ' ', c.last_name) AS
Клієнт,
                CONCAT(e.first_name, ' ', e.last_name) AS Працівник,
                o.order_date AS Дата, o.status AS Статус, o.total_amount AS
Сума
        FROM orders o
        JOIN customers c ON o.customer_id = c.customer_id
        JOIN employees e ON o.employee_id = e.employee_id
        WHERE c.first_name LIKE %s OR c.last_name LIKE %s OR e.first_name
LIKE %s OR o.status LIKE %s
        ORDER BY o.order_date DESC
    "",
    "customers": ""
        SELECT customer_id AS ID, first_name AS "Ім'я", last_name AS
Прізвище,
                phone AS Телефон, email AS Email, address AS Адреса
        FROM customers
        WHERE first_name LIKE %s OR last_name LIKE %s OR phone LIKE %s OR
email LIKE %s
        ORDER BY customer_id
    "",
    "employees": ""
        SELECT employee_id AS ID, first_name AS "Ім'я", last_name AS
Прізвище,
                position AS Посада, phone AS Телефон, salary AS Зарплата
        FROM employees
        WHERE first_name LIKE %s OR last_name LIKE %s OR position LIKE %s
OR phone LIKE %s
        ORDER BY employee_id
    "",
    "suppliers": ""
        SELECT supplier_id AS ID, supplier_name AS Постачальник,
                phone AS Телефон, email AS Email, address AS Адреса
        FROM suppliers
        WHERE supplier_name LIKE %s OR phone LIKE %s OR email LIKE %s OR
address LIKE %s

```

```

        ORDER BY supplier_id
        """
    }

    if self.current not in search_sql:
        self.show(self.current)
        return

    cols, rows = self.query(search_sql[self.current], (like, like, like, like))
    self.fill("Результати пошуку", cols, rows)

def add_customer_window(self):
    fields = [("Ім'я", "first_name"), ("Прізвище", "last_name"),
              ("Телефон", "phone"), ("Email", "email"), ("Адреса", "address")]
    self.form_window("Додати клієнта", fields, self.save_customer)

def order_window(self):
    fields = [("ID клієнта", "customer_id"), ("ID працівника", "employee_id"),
              ("ID товару", "product_id"), ("Кількість", "quantity")]
    self.form_window("Створити замовлення", fields, self.save_order)

def form_window(self, title, fields, save_func):
    win = tk.Toplevel(self.root)
    win.title(title)
    win.geometry("460x380")
    win.configure(bg="#eef3f8")
    win.resizable(False, False)

    tk.Label(win, text=title, bg="#eef3f8", fg="#1f2937",
              font=("Segoe UI", 16, "bold")).pack(pady=15)

    form = tk.Frame(win, bg="white", highlightbackground="#d1d5db",
                    highlightthickness=1)
    form.pack(fill="both", expand=True, padx=20, pady=10)

    entries = {}

    for i, (label, key) in enumerate(fields):
        tk.Label(form, text=label, bg="white", font=("Segoe UI", 10,
"bold")).grid(
            row=i, column=0, padx=15, pady=10, sticky="w"
        )
        ent = tk.Entry(form, font=("Segoe UI", 11), bg="#f9fafb", relief="solid",
bd=1)
        ent.grid(row=i, column=1, padx=15, pady=10, ipady=4)
        entries[key] = ent

        self.button(form, "Зберегти", lambda: save_func(win, entries),
"#16a34a").grid(
            row=len(fields), column=0, colspan=2, pady=20
        )

def save_customer(self, win, e):
    data = {k: v.get().strip() for k, v in e.items()}

    if not data["first_name"] or not data["last_name"] or not data["phone"]:
        messagebox.showwarning("Помилка", "Заповніть ім'я, прізвище та телефон")
        return

    self.query("""

```

```

        INSERT INTO customers(first_name, last_name, phone, email, address)
        VALUES(%s, %s, %s, %s, %s)
        """ , tuple(data.values()), fetch=False)

messagebox.showinfo("Успіх", "Клієнта додано")
win.destroy()
self.show("customers")

def save_order(self, win, e):
    try:
        customer_id = int(e["customer_id"].get())
        employee_id = int(e["employee_id"].get())
        product_id = int(e["product_id"].get())
        quantity = int(e["quantity"].get())

        if quantity <= 0:
            raise ValueError
    except ValueError:
        messagebox.showwarning("Помилка", "ID та кількість повинні бути
        правильними числами")
        return

    con = self.connect()
    if not con:
        return

    try:
        cur = con.cursor()

        cur.execute("SELECT price, quantity_in_stock FROM products WHERE
        product_id=%s", (product_id,))
        product = cur.fetchone()

        if not product:
            messagebox.showwarning("Помилка", "Товар не знайдено")
            return

        price, stock = product

        if quantity > stock:
            messagebox.showwarning("Помилка", "Недостатня кількість товару")
            return

        cur.execute("""
            INSERT INTO orders(customer_id, employee_id, status, total_amount)
            VALUES(%s, %s, 'нове', %s)
            """, (customer_id, employee_id, price * quantity))

        order_id = cur.lastrowid

        cur.execute("""
            INSERT INTO order_items(order_id, product_id, quantity, unit_price)
            VALUES(%s, %s, %s, %s)
            """, (order_id, product_id, quantity, price))

        con.commit()
        messagebox.showinfo("Успіх", f"Замовлення №{order_id} створено")
        win.destroy()
        self.show("orders")

```

```
        except Error as err:
            con.rollback()
            messagebox.showerror("Помилка SQL", str(err))
        finally:
            con.close()

if __name__ == "__main__":
    root = tk.Tk()
    ShopApp(root)
    root.mainloop()
```

ДОДАТОК В  
(обов'язковий)

**ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ**

Хмельницький національний університет  
Кафедра інженерії програмного  
забезпечення

Кваліфікаційна робота на тему:

Вебсайт онлайн-галереї з  
продажу картин

Здобувач: Максим Божик

Керівник: канд. техн. наук, доцент Юрій Форкун

Рисунок В.1 – слайд «Титульна сторінка»

# Актуальність теми

- ✓ Електронна комерція є одним із ключових напрямів сучасного бізнесу.
- ✓ Інтернет-магазини забезпечують швидкий доступ до товарів та послуг.
- ✓ Продаж картин через вебресурс дозволяє автоматизувати процес замовлення.
- ✓ Використання сучасних вебтехнологій підвищує зручність для користувачів.

Рисунок В.2 – слайд «Актуальність теми»

# Мета та завдання

- ✓ Мета: проектування та розробка вебсайту для продажу картин.
- ✓ Розробка структури сайту.
- ✓ Проектування архітектури та інтерфейсу.
- ✓ Реалізація функціоналу інтернет-магазину.
- ✓ Розробка вебінтерфейсу та каталогу товарів.
- ✓ Тестування програмного забезпечення.

Рисунок Е.3 – слайд «Мета та завдання»

# Аналіз предметної області

- ✓ Досліджено сучасні інтернет-магазини та сервіси електронної комерції.
- ✓ Визначено основні функції: каталог, кошик, авторизація, оплата.
- ✓ Розглянуто роль пошуку, рекомендацій та персоналізації.
- ✓ Визначено структурні та функціональні особливості системи.

Рисунок В.4 – слайд «Змістовний аналіз предметної області, її структурних та функціональних особливостей»

# Аналіз програмно-технічного забезпечення

<u>Система</u>	<u>Переваги</u>	<u>Недоліки</u>	<u>Пошук</u>
ArtSale.ua	<u>Зручний функціонал</u>	Обмежений дизайн	Так
art-holst.com.ua	Мінімалістичний інтерфейс	Мало функцій	Так
JoseArtGallery	Великий каталог	Багато спливаючих вікон	<u>Ні</u>

Рисунок В.5 – слайд «Аналіз наявного програмно-технічного забезпечення»

# Функціональні та нефункціональні ВИМОГИ

- ✓ Авторизація та реєстрація користувачів.
- ✓ Перегляд каталогу картин та категорій.
- ✓ Кошик та оформлення замовлення.
- ✓ Захист даних користувачів.
- ✓ Масштабованість та зручність використання.
- ✓ Виконання принципу «трьох кліків».

Рисунок В.6 – слайд «Визначення функціональних та нефункціональних вимог до ПЗ»

# Архітектура та шаблони проектування

- ✓ Обрано клієнт-серверну архітектуру.
- ✓ Використано шаблон MVC.
- ✓ Розділення на Model, View, Controller.
- ✓ Підвищення масштабованості та підтримуваності системи.

Рисунок В.7 – слайд «Вибір типу архітектури та шаблонів проектування»

# Декомпозиція та інтерфейси

- ✓ Виділено модулі користувача та адміністратора.
- ✓ Реалізовано взаємодію клієнта із сервером.
- ✓ Передбачено REST-подібну взаємодію.
- ✓ Реалізовано модулі авторизації та керування товарами.

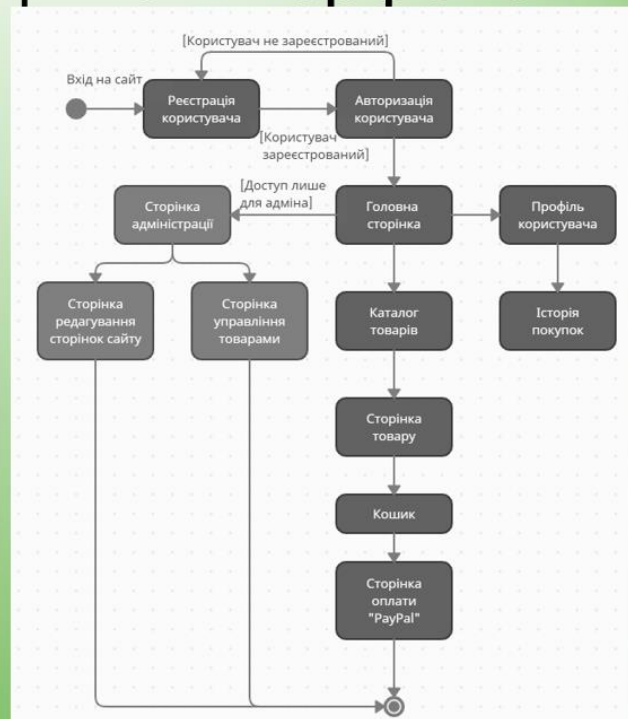


Рисунок В.8 – слайд «Опис декомпозиції, залежностей, інтерфейсів»

# Аналіз та вибір технологій

- ✓ Мова програмування: C#.
- ✓ Платформа: ASP.NET Core MVC.
- ✓ СУБД: Microsoft SQL Server.
- ✓ Середовище розробки: Microsoft Visual Studio.
- ✓ Технології обрані через масштабованість та підтримку MVC.

Рисунок В.9 – слайд «Аналіз та вибір технологій»

# Проектування модулів і даних

- Спроектовано базу даних із 8 таблиць із основними сутностями.
  - Users, Roles, Products, Categories, Orders
- Реалізовано зв'язки між сутностями.
- Здійснено нормалізацію до 3НФ.

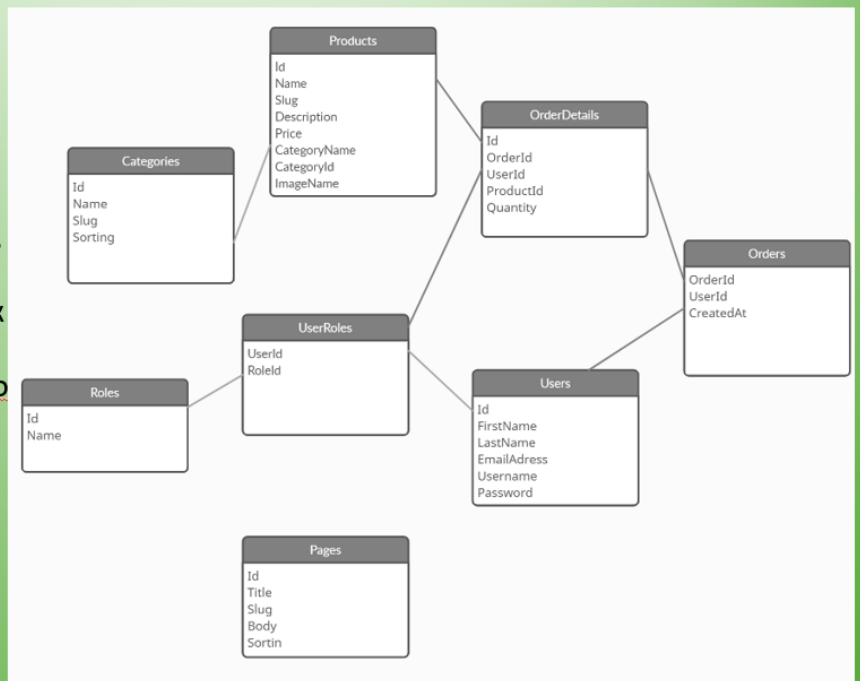


Рисунок В.10 – слайд «Проектування БД»

# Програмна реалізація

- ✓ Реалізовано базу даних та модулі системи.
- ✓ Створено каталог картин та кошик.
- ✓ Реалізовано сторінки авторизації та реєстрації.
- ✓ Створено адміністративну панель.

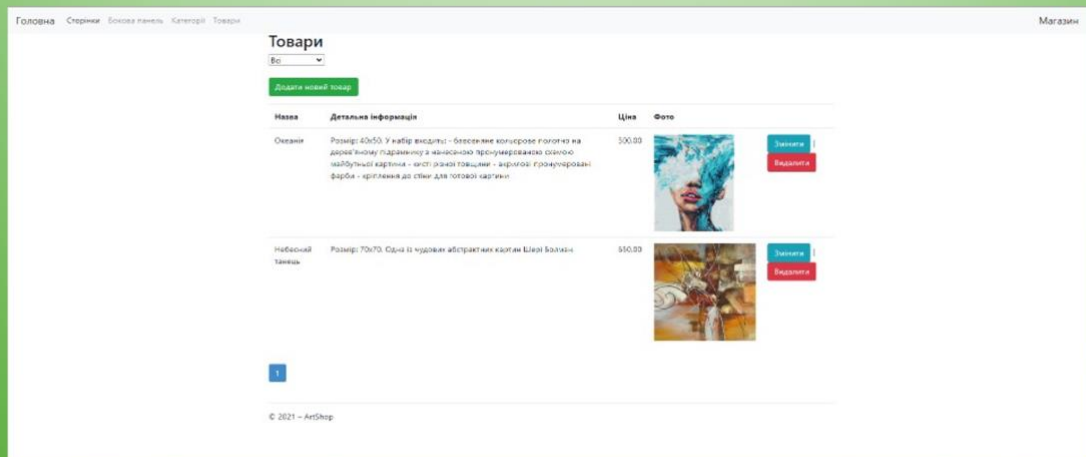


Рисунок В.11 – слайд «Програмна реалізація»

# Тестування програмного забезпечення

Під час тестування було виявлено та усунено 2 критичні помилки.

Проблема	Результат	Статус
Некоректне відображення товарів	Виправлено структуру БД	Усунено
Помилка підрахунку вартості	Виправлено алгоритм	Усунено
Орфографічні помилки	Оновлено контент	Усунено

Рисунок В.12 – слайд «Тестування ПЗ»

# Висновки

Завдання	Результат
Розробка структури сайту	Створено структуру інтернет-магазину
Проектування сайту	Розроблено архітектуру MVC
Реалізація сайту	Створено функціональний вебресурс
Розробка вебінтерфейсу	Створено UI для користувачів
Тестування ПЗ	Виявлено та усунено помилки

Результатом роботи став функціональний інтернет-магазин з продажу картин.

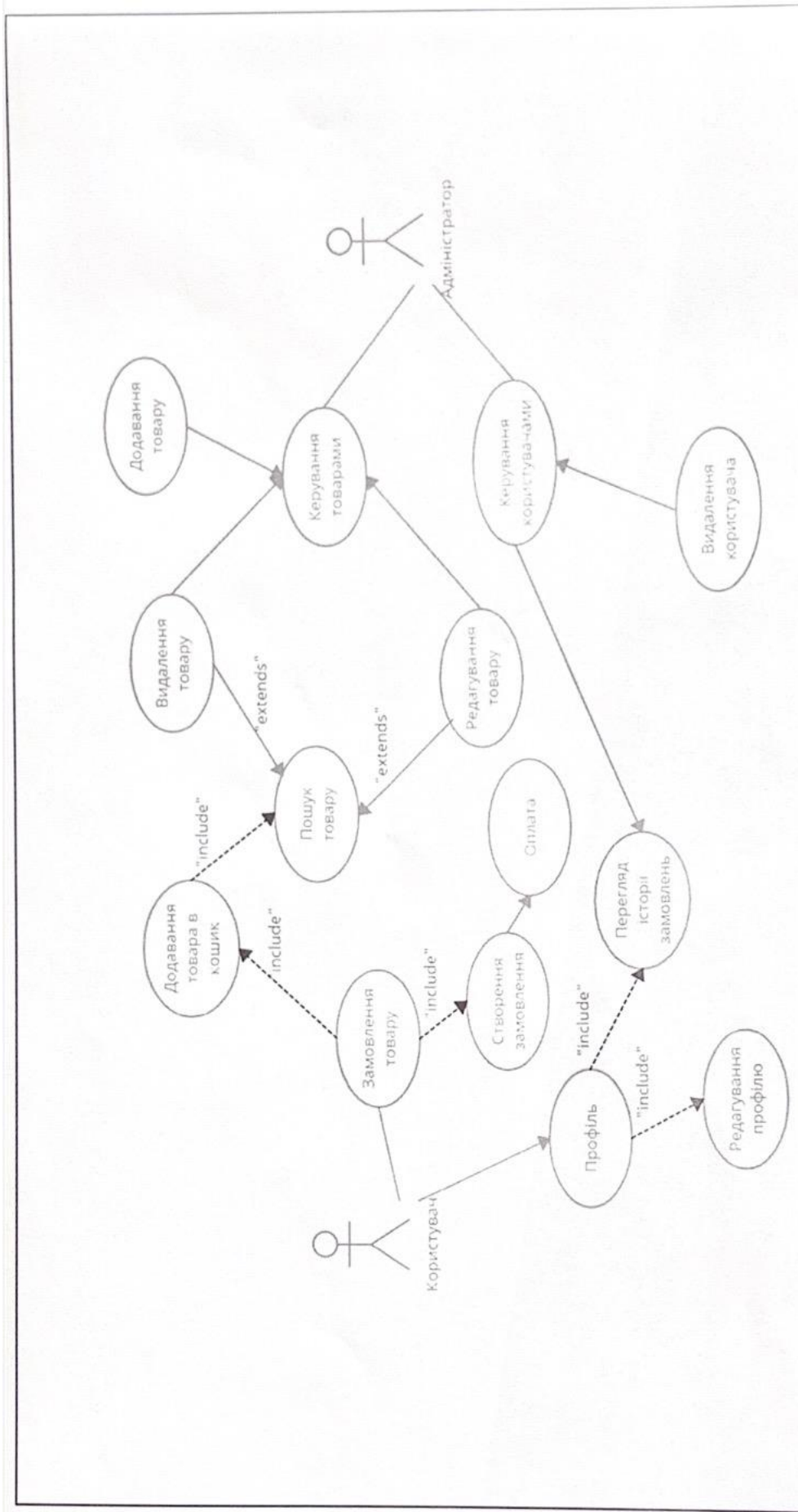
Рисунок В.13 – слайд «Висновки»

**Дякую за увагу!**

Готовий відповісти на запитання.

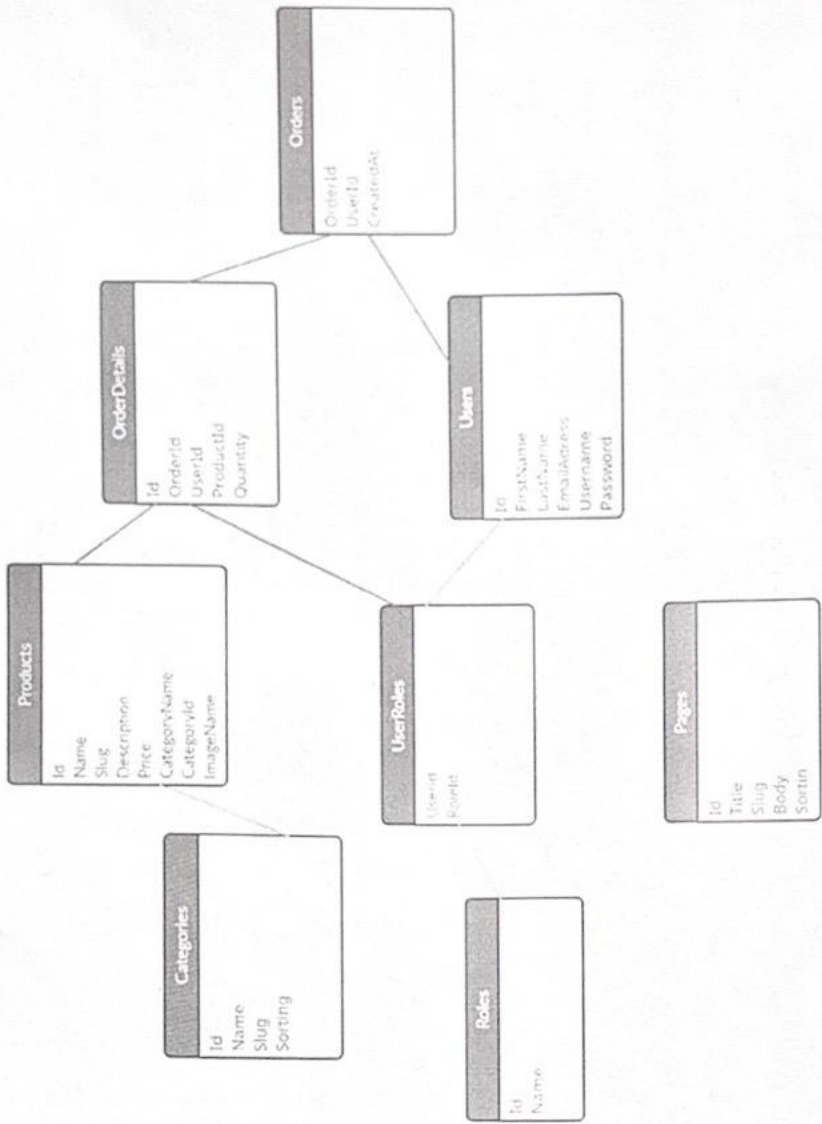
Рисунок В.14 – слайд «Дякую за увагу»

# **Графічні матеріали**



КерІПЗ.220186.01.03.Г8

Зміст	Лист	№ докум	Поліс	Дат	Літ.	Маса	Масштаб
Розробка		Божик М.С.		2018			
Керівник		Форкун Ю. В.		2018			
Н. Контроль		Яшина О. М.					
Діаграма варіантів використання							
Арк. 11 Аркуші 3							
ХНУ, ІПЗ-22-1							



КерІПЗ.220196.01.03.Г8

Звич. Пункт	№ докум	Гречис. Дати	Лит.	Маса	Масштаб
Розробив	Божик М.С.	11.08.2022			
Керівник	Фурчук Ю. В.	11.08.2022			
Автори	Яворний О. М.				
Схема бази даних					
АДК: 1 АДКУШІ 3					
ХНУ, ІПЗ-22-1					

Завідувачу кафедри  
інженерії програмного забезпечення  
проф. Бедратюку Л. П.  
студента групи ІПЗ-21-1

Божик М. С.

Прізвище, ініціали

### ЗАЯВА

Прошу закріпити за мною тему кваліфікаційної роботи освітнього ступеня  
«бакалавр» за спеціальністю 121 «Інженерія програмного забезпечення»: Вебсайт  
онлайн-галереї з продажу картин

(керівник роботи –



Форкун Юрій Вікторович)

Прізвище, ім'я, по батькові

28.05.2025

Дата



Підпис студента

Завідувачу кафедри інженерії програмного  
забезпечення проф. Леоніду БЕДРАТЮКУ  
здобувача вищої освіти

Божика Максима Сергійовича  
факультет ІТ, IVкурс, група ІПЗ-22-1

### ЗАЯВА

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

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

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

1.05.26

дата

  
підпис

# Anti-Plagiarism (<http://ap.km.ua>) v-16.718

Максимальне співпадіння з одним документом 11.0%

Словники перевірки: UA, US, RU. Помилки в документах: 11%

ID: 273290 Назва: БКР Вебсайт онлайн-галереї з продажу картин Долано в БД: 2026-06-03 Автора: Максим БОЖИК Керівники: канд. техн. наук, доцент Юрій ФОРКУН Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	89710	769	15061 (17%)	174 (23%)

## Джерело платігу

ID	Опис	Наявність платігу в документі	
		Символи	Лексеми
272082	Назва: Мобільний застосунок для обліку домашньої аптечки з функцією нагадувань Долано в БД: 2026-05-25 Автора: Тарісія ПИШКА Керівники: канд. техн. наук, доцент Оксана ЯШИНА Консультанти: Опоненти:	9539 (11.0%)	100 (13.0%)

Протокол аналізу звіту подібності науковим керівником

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

**Автор:** Максим БОЖИК

**Співавтор:**

**Назва:** Вебсайт онлайн-галереї з продажу картин

**Науковий керівник:** канд. техн. наук, доцент Юрій ФОРКУН

**Підрозділ:** Кафедра інженерії програмного забезпечення

**Коефіцієнт подібності 1:** 6.56%

**Коефіцієнт подібності 2:** 1.77%

**Мікропробіли:** 43

**Заміна букв:** 2

**Інтервали:** 0

**Білі знаки:** 0

**Дата створення звіту:** 2026-06-03 13:46:04.0

Після аналізу Звіту подібності констатую наступне:

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

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

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

Обґрунтування:

Дата

03.06.26

експерт

*Юрій Форкун* (Юрій ФОРКУН)

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ  
освітнього ступеня «Бакалавр»

Дипломник Божик Максим Сергійович

Тема Вебсайт онлайн-галереї з продажу картин

Спеціальність 121 – Інженерія програмного забезпечення

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3; кількість сторінок записки 82

1. Короткий зміст пояснювальної записки та прийнятих рішень У кваліфікаційній роботі проаналізовано предметну область онлайн-освіти та обгрунтовано доцільність створення системи персоналізованих рекомендацій. Визначено вимоги до системи, спроектовано архітектуру на основі Microsoft SQL Server та технології Entity Framework Core та створено клієнтську частину з використанням HTML5, CSS3, JavaScript та Bootstrap. Проведене тестування засвідчило коректність роботи системи та відповідність результатів очікуваним метрикам.

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

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки та передових методів роботи У вступі обгрунтовано актуальність теми, сформульовано мету та завдання роботи. У першому розділі проаналізовано предметну область онлайн-освіти, класифіковано підходи до персоналізації та визначено вимоги до системи. У другому розділі розглянуто архітектурні рішення, інформаційну модель і технологічний стек, на основі яких спроектовано компонентну структуру системи. У третьому розділі описано реалізацію програмних модулів, включно з генерацією даних, навчанням моделі та тестуванням за метриками. За підсумками тестування підтверджено функціональну придатність системи та відповідність її поведінки заданим вимогам

4. Позитивні сторони роботи Обрана тема є актуальною в контексті зростання попиту на персоналізовані освітні рішення. У роботі обгрунтовано вибір сучасної технологічної зв'язки, яка відповідає вимогам масштабованості, гнучкості та продуктивності. Реалізація системи ґрунтується на актуальних практиках розробки рекомендаційних сервісів, включаючи гібридну модель, обробку даних у форматі JSON, що свідчить про високий рівень технічної підготовки автора.

5. Негативні сторони роботи Реалізація охоплює лише базову функціональність системи; не передбачено інтеграції з реальними платформами, тестування або розширеної аналітики користувацької поведінки.

6. Оцінка графічного оформлення та пояснювальної записки Графічне оформлення відповідає тематичі роботи: діаграми, таблиці та рисунки логічно доповнюють текст і пояснюють архітектуру, інформаційну модель та алгоритмічні підходи. Пояснювальна записка оформлена відповідно до чинних стандартів, структура витримана, зміст подано послідовно й обґрунтовано.

7. Відгук про кваліфікаційну роботу в цілому Робота виконана на високому рівні та заслуговує на позитивну оцінку. Пояснювальна записка логічно структурована, зміст викладено чітко й послідовно, що сприяє легкому сприйняттю технічного матеріалу. Графічні матеріали доречно ілюструють ключові етапи проєктування системи, зокрема архітектуру, логіку обробки даних і сценарії взаємодії, посилюючи загальне враження цілісності та завершеності розробки.

8. Інші зауваження \_\_\_\_\_

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

РЕЦЕНЗЕНТ (прізвище, ім'я, по-батькові, посада, місце роботи)

Головко Ольга Олександрівна, д.ф.рочка,  
зав. каф. ІИС

“ 8 ”

06

2025 р.

  
(підпис)

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

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

Назва кваліфікаційної роботи: «Вебсайт онлайн-галереї з продажу картин»

Автор: Божик Максим Сергійович

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

Спеціальність: 121 – Інженерія програмного забезпечення

Науковий керівник: Форкун Юрій Вікторович, кандидат технічних наук, доцент

Після аналізу звіту/звітів зроблено такий висновок:

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

Підтвердження:

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

- 1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 2) система фіксувала технічні особливості (наприклад, поєднання латиниці й українських індексів), а не модифікацію тексту.

Сумарний обсяг запозичень — 6.56%, що відповідає науковим стандартам і не впливає на якість кваліфікаційної роботи.

Дата 5.06.2026

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи



Леонід БЕДРАТІЮК

Леонід БЕДРАТІЮК

Юрій ФОРКУН